

**VŠB – Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra informatiky**

**Doporučování informací v mobilních zařízeních**  
**Information Recommendation on Mobile Devices**

**Květen 2013**

**Bc. Martin Vichnar**

VŠB - Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

## Zadání diplomové práce

Student: **Bc. Martin Vichnar**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: **Doporučování informací v mobilních zařízeních**  
**Information Recommendation on Mobile Devices**

### Zásady pro vypracování:

Množství informací, které máme k dispozici, stále exponenciálně roste. Jde nejen o informace na "klasickém" webu, ale i v sociálních či profesních sítích. Proto automatické doporučení (filtrování) informací se stává klíčovým aspektem práce s informacemi. Tento proces významně ovlivňuje i kontext získávání informací, který zahrnuje použité zařízení. Zpřístupňování informací na mobilních zařízeních má několik specifika, která vyplývají zejména z velikosti displeje zařízení a způsobu práce s ním (dokážeme poměrně přesně zjistit, zda uživatel pracuje nebo nepracuje s mobilním zařízením a tedy dokážeme získat implicitní zpětnou vazbu bez toho, abychom uživatele speciálně sledovali).

Otevřenými problémy v této oblasti jsou např. odhadování charakteristik a zájmů uživatele na základě jeho aktivity, odhalování vztahů mezi lidmi na základě jejich chování a následnou využití pro efektivnější doporučení, personalizovaná navigace v informačním prostoru (např. tok informací v sociálních sítích Facebook nebo Google +, novinové zprávy), identifikace autorit a doporučení na základě nich nebo kolaborativní značkování informačních zdrojů na mobilním zařízení a následné využití značek pro doporučení informací.

Je třeba se zaměřit na některou z oblastí a navrhnout a experimentálně ověřit metodu, která bude přispěvkem k doporučení informací v mobilních zařízeních. Ověření metody se může vázat na aplikační doménu výuky programování s využitím adaptivního systému XAPOS, ale i práci s digitálními knihovnami, skupinovým vývojem softwaru, zprávami a podobně.

### Seznam doporučené odborné literatury:

- [1] Kruger, J. Baus, D. Heckmann, M. Kruppa, and R. Wasinger. Adaptive Mobile Guides. In P. Brusilovsky et al. (Eds.): The Adaptive Web, LNCS 4321, Springer-Verlag Berlin Heidelberg. pp. 521–549, 2007.
- [2] P. Pombinho. Information visualization on mobile environments. In Proc. of the 12th Int. Conf. on Human computer interaction with mobile devices and services (MobileHCI '10). ACM, New York, NY, USA, 493-494, 201
- [3] Šaloun P., Velart Z., Concept Space Rating for Personalization of Learning Materials Based on Relations. In. Proc. of SMAP'09, 4th International Workshop on Semantic Media Adaptation and Personalization, CS IEEE 2009, pp. 67-72 (2009).
- [4] Domácí stránka systému XAPOS: <http://arg.vsb.cz/XAPOS/>

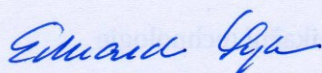


Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. RNDr. Petr Šaloun, Ph.D.**

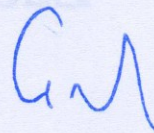
Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2013



doc. Dr. Ing. Eduard Sojka  
vedoucí katedry





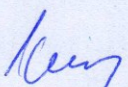
prof. RNDr. Václav Snášel, CSc.  
děkan fakulty



## **Prohlášení studenta**

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Dne: 2.5.2013

  
.....  
podpis studenta

## **Poděkování**

Rád bych poděkoval panu doc. RNDr. Petru Šalounovi, Ph.D, za odbornou pomoc a konzultaci při vytváření této diplomové práce.

## **Abstrakt**

Diplomová práce se zabývá problematikou doporučování informací v mobilních zařízeních, aplikační doménou je cestovní ruch. Práce obsahuje přehled cestovního ruchu a analyzuje možnosti mobilních zařízení s cílem navrhnout a realizovat transformaci existujícího webového informačního systému do rozšířené verze vhodné pro mobilní zařízení. Práce zahrnuje poznatky z informačních systémů, webového a sociálního inženýrství a také z oblasti marketingu - je uplatněn mezioborový přístup a výsledkem je prototyp aplikace určené k doporučování zájezdů na mobilních zařízeních (dále zkráceně TARS). Při vývoji TARS byla pro doporučování zájezdů vybrána metoda CBR, která na základě naučených případů a metrik podobnosti nabídne uživateli seznam doporučení. Ukázky rozhraní a testovacího provozu jsou v práci uvedeny.

## **Klíčová slova**

Mobilní zařízení, doporučování informací, cestovní ruch, CBR, konverzační metody, metriky podobnosti, personalizace

## **Abstract**

Thesis deals with information recommending in mobile devices, application domain is travel industry. Thesis contains summary of travel industry and analyzes mobile device's possibilities with aim to design and implement transformation of existing internet information system into extended version more suitable for mobile devices. Thesis contains knowledge of information systems, web and social Engineering and also marketing. Interdisciplinary approach is used to develop application prototype for holiday packages recommendation on mobile devices (hereinafter TARS). CBR method was selected in TARS development for recommendation of holiday packages, as it offers user a list of recommendations based on previously learned cases and metrics of similarities. Examples of GUI and alfa testing are listed in thesis.

## **Keywords**

Mobile devices, information recommendation, travel industry, CBR, conversation techniques, similarity metrics, personalization

## Seznam použitých zkratek

**CK** – cestovní kancelář

**CA** – cestovní agentura

**HW** – hardware

**SW** – software

**OC** – OfferCase

**DC** – DialogCase

**CBR** – Case Based Reasoning – Rozhodování založené na případech

**SBR** – Similarity Based Recommendation – Doporučení založené na podobnosti

**CPU** – procesor

**TARS** – Travel Agency Recommender System

**FBR** – Filter Based Recommendation – Doporučování na základě přesné shody

# Obsah

1	Úvod.....	10
2	Mobilní zařízení .....	11
2.1	Chytrý telefon.....	11
2.1.1	Omezení a přednosti.....	11
2.1.2	Zobrazování informací .....	11
2.1.3	Ovládání .....	12
2.1.4	Připojení .....	12
2.1.5	Výkon.....	13
2.1.6	Shrnutí.....	13
2.2	Výběr platformy, výhody, nevýhody, přínosy .....	13
2.2.1	Plnohodnotná nativní aplikace .....	13
2.2.2	Aplikace ve formě HTML, CSS s podporou javascriptu .....	15
2.2.3	Hybridní aplikace .....	15
2.3	Shrnutí.....	16
3	Cestovní ruch – úvod do problematiky.....	17
3.1	Zájezd, termín a služby .....	17
3.2	Agregace zájezdů .....	18
3.3	Současný stav a konkurence.....	18
3.4	Nabídka zájezdů – <a href="http://www.zajezdyonline.cz">www.zajezdyonline.cz</a> .....	19
3.4.1	Hlavní strana .....	19
3.4.2	Vyhledávač zájezdů.....	20
3.4.3	Nabídka výsledků vyhledávání .....	21
3.4.4	Detail zájezdu.....	21
3.5	Shrnutí.....	22
4	Doporučování informací.....	23
4.1	Základní dělení.....	23
4.1.1	Kolaborativní metody - Collaborative based.....	24
4.1.2	Metody založené na obsahu - Content based .....	25
4.1.3	Metody založené na znalostech - Knowledge based .....	25
4.1.4	Výběr metody .....	26
4.2	Case-based doporučení .....	27



4.2.1	Rozhodování založené na případech - Case Based Reasoning.....	27
4.2.2	Posuzování podobnosti.....	28
4.2.3	Reprezentace CBR .....	29
4.3	Single-shot problém vs. konverzační metody.....	30
4.3.1	Metoda dotazování .....	31
4.3.2	Metoda průběžných doporučení .....	33
4.3.3	Hybridní řešení.....	34
4.4	Personalizace .....	34
4.4.1	Kontext.....	34
4.4.2	Profilování .....	34
4.5	Shrnutí.....	36
5	Aplikace doporučování zájezdů - TARS.....	37
5.1	Architektura.....	37
5.2	Část dialogová.....	38
5.2.1	Zvládnutelný set.....	39
5.2.2	Nezvládnutelný pod-specifikovaný set .....	39
5.2.3	Nezvládnutelný přespecifikovaný set.....	40
5.2.4	Dialog case – DC.....	42
5.2.5	Profilování.....	42
5.3	Část doporučení .....	43
5.3.1	Proces učení vah.....	44
5.3.2	Offer case – OC.....	45
5.4	Posuzování podobnosti .....	46
5.4.1	Binární vlastnosti.....	46
5.4.2	Pole binárních vlastností .....	46
5.4.3	Číselné vlastnosti.....	47
5.4.4	Datumové vlastnosti.....	48
5.4.5	Taxonomie.....	48
5.4.6	Výčtové vlastnosti .....	49
5.4.7	Normalizace .....	49
5.5	GUI.....	49
5.5.1	Zadání požadavku a dialog.....	50
5.5.2	Prezentace výsledků doporučení .....	50
5.5.3	Prezentace nabídky – detail zájezdu.....	52
5.5.4	Reprezentace případu CBR .....	53

5.5.5	Reprezentace vlastností a míry podobnosti.....	53
6	Nastavení a výsledky .....	54
6.1	Nastavení.....	54
6.1.1	Nastavení prahových hodnot zvládnutelného setu .....	54
6.1.2	Nastavení koeficientu inkrementace/dekrementace .....	54
6.2	Výsledky.....	54
6.3	Statistiky.....	55
7	Závěr.....	57

# 1 Úvod

Má diplomová práce se zabývá problematikou doporučování informací v mobilních zařízeních.

Aplikační doménou je oblast cestovního ruchu, která je specifická množstvím a nesourodostí nabízených informací a složitým uživatelským rozhraním uživatelských aplikací, jehož transformace do mobilního zařazení si ve svém důsledku vyžaduje mezioborový přístup na pomezí informačních systémů, webového a sociálního inženýrství, doporučování i marketingu s těsnou vazbou na mobilní technologie.

Cílem diplomové práce je navrhnout metodu pro doporučování informací v mobilních zařízeních a experimentálně ji ověřit na prototypové aplikaci v doméně prodeje zájezdů.

Práce je rozdělena do sedmi kapitol a je doplněna jednou přílohou. Po úvodu následuje kapitola 2, která popisuje specifika mobilních zařízení, jejich omezení a možnosti. Podrobněji se věnuje výběru platformy vhodné pro implementaci prototypového systému. Kapitola 3 je věnována problémové doméně cestovního ruchu – prodeji zájezdů. Představena je také existující aplikace pro PC jako nemobilní zařízení, diskutovány jsou problémy, které je nutné řešit na mobilní platformě. V kapitole 4 jsou rozebírány techniky doporučování informací a produktů. Vybrána a blíže popsána je metoda CBR (Case Based Reasoning – rozhodování založené na případech), techniky dialogu a učících algoritmů. V kapitole 5 jsou poznatky předchozích kapitol vhodně užity v prototypové aplikaci TARS (Travel Agency Recommender System). Postupy a algoritmy doporučování a dialogu jsou prezentovány na konkrétních případech projektu TARS. Kapitola 6 prezentuje výsledky testování úspěšnosti doporučení a ukazuje postupy nastavení. Práce je stručně shrnuta v kapitole 7.



## 2 Mobilní zařízení

Mobilní zařízení je označováno také jako kapesní zařízení do ruky nebo kapesní počítač. Jde o přenosné zařízení zpravidla s nestálým datovým spojením. Je vybaveno displejem, obvykle dotykovým, někdy i klávesnicí různých velikostí.

Dnes si mobilní zařízení nejvíce lidí spojí asi s chytrými telefony, případně tablety. Mezi ně patří určitě i různé čtečky, navigace, PDA, klasické mobilní telefony, různá speciální podniková mobilní zařízení atp. Pro naše potřeby bude ale předmětem primárního zájmu právě *chytrý telefon*, případně *tablet*. Ten ale s trochou nadsázky můžeme považovat za chytrý telefon s větším displejem.

### 2.1 Chytrý telefon

Chytrý telefon neboli smartphone je telefon, který je založen na pokročilém operačním systému s aplikačním rozhraním s možností instalace software. Vyznačuje se také tím, že je zpravidla připojen k internetu. Podrobná definice je k dispozici na pdaSoft [1].

Přestože je přímo v názvu telefon, postupem času se telefonování stalo jen jednou z mnoha funkcionalit, které smartphone nabízí. Mezi ty hlavní můžeme řadit prohlížeč webových stránek, navigace, přehrávání multimédií, emailový klient, fotoaparát, videokamera a další.

#### 2.1.1 Omezení a přednosti

Nespornou výhodou a rozdílem proti klasickému PC je jeho přenositelnost. Nebo lépe, "mobilita bez přívlastků". Protože jej můžeme mít stále u sebe, slouží nejen jako komunikační prostředek, ale také jako společník pro krácení volných chvil, jako je čekání na úřadech, u lékaře, na dopravní prostředek, ale i přímo při přepravě v MHD atp. Právě toto je prostor pro mnohé aplikace nebo služby a možnost pro firmy nabízet svůj katalog s výrobky nebo službami zákazníkovi. Ten jinak třeba nemá dostatek času si nabídku prohlédnout. Smartphone je tedy ideální platformou pro obdobné aktivity. Sice má jistá omezení, ale minimálně pro před-výběr produktu naprosto dostačuje.

#### 2.1.2 Zobrazování informací

Všechna mobilní zařízení disponují displejem. Byly sice i tendence zobrazování na principu projekce, ale v běžném prodeji jsem se s nimi nesetkal. Protože jde o „mobilní“ zařízení, už z principu disponuje přenosným a tedy nevelkým displejem. Rozlišení sice neustále roste, dokonce se objevují přístroje s full HD rozlišením, ale s velikostí displeje cirká 4 palce bude mít asi při maximálním využití rozlišení většina lidí problém vůbec něco přečíst. U tabletů je to obdobné - rozlišení roste, ale spíš už jen pro „zneviditelnění“ pixelů a přirozenější zobrazení. Máme tedy omezení ve velikosti ovládacích prvků na obrazovce, které je více závislé na velikosti displeje než na množství obrazových bodů.

Druhým hendikepem mobilních zařízení je čitelnost na slunci. Existují sice e-ink displeje<sup>1</sup>, ale pro jejich přílišnou pomalost a slabou barevnost jsou například pro prohlížení webových stránek jen málo využitelné. Je tedy dobré si dát pozor při zobrazování informací na dobré kontrastování barev a

---

<sup>1</sup> <http://www.ecti.cz/eink-technologie/>

více upřednostnit informace před vzhledem. Do jisté míry je řešením možnost přepínat denní a noční barvy jako to umožnili u Sunnysoftu<sup>2</sup>.

### 2.1.3 Ovládání

Starší chytré telefony byly často vybavovány hardwarovou klávesnicí. Postupně se klávesnice zmenšovala přes numerickou po dnešních nezbytných několik hw kláves. Zbytek se přesouvá na dotykovou klávesnici. Dokonce se již objevují tendence dotykových displejů s hw odezvou<sup>3</sup>, prozatím ale jen jako hudba budoucnosti.

Dá se ale předpokládat, že ovládání se bude stále více zaměřovat na dotykový displej ve spolupráci s hlasovým ovládáním. Přes všechny výhody přímého zadávání povelů na displeji ale vyvstávají na vývojáře jisté požadavky a omezení. Vzhledem k tomu, že mobilní zařízení většinou disponuje malým displejem, je nutné brát ohledy na velikost ovládacích prvků, vzhledem k velikosti prstů a jejich přesnosti. To co je možné u klasického PC ovládaného myší, může být na smartphone problémem. Prohlížení webových stránek původně určených pro velké monitory, se pak stává dosti nepohodlné. Jsme nuceni neustále posunovat zvětšenou část po displeji a ztrácíme přehled o celkovém rozložení stránky. Internetové prohlížeče se sice snaží práci usnadnit, případně existují konvertory stránek [2], které upraví „velkou“ webovou stránku na „malou“, ale velmi záleží jak je stránka naprogramována nebo i jestli je přímo uzpůsobena pro konvertování. Nejjistější je varianta přepracování webové stránky pro smartphone. Nativní aplikace jsou kapitolou samy o sobě. Představa použití aplikace přímo z velkého PC je téměř nemožná. Vždy musí být brán ohled na ovládání pomocí prstů na malém dotykovém displeji.

### 2.1.4 Připojení

Propojení zařízení s okolním světem je nutností. Pro jednoduchost se budeme bavit jen o datových spojeních. Historicky byla využívána vytáčená spojení (CSD), případně stále aktuální technologie postavené na GSM sítích (GPRS, EDGE). V dnešní době již mluvíme o nových generacích sítí (3G, LTE), které přináší mnohem vyšší přenosové rychlosti (více o datových přenosech a použitých zkratkách v [3]).

V době psaní textu diplomové práce došlo na českém trhu k zásadním úpravám cen a limitů datových tarifů, které se tak staly dostupnějšími. Můžeme tedy v budoucnu předpokládat, že každý uživatel mobilního zařízení disponuje trvalým připojením k internetu. To je důležité pro zamýšlenou aplikaci, určenou pro výběr zájezdů s častými změnami cen, kde budeme vyžadovat aktuální data.

Přes příznivý vývoj v rychlostech a dostupnosti spojení, je dobré dbát na velikost přenášených dat – obrázků a knihoven.

Otázkou zůstává, co nabídnout zákazníkovi ve chvíli, kdy připojení ztratí, kdy například v dopravním prostředku nemusí být spojení ideální po celou dobu. A právě lidé v dopravních prostředcích jsou ideálními adepty na využití zamýšlené aplikace. V další verzi aplikace by stálo za úvahu například nabídnout v době výpadku klientovi zájezdy, které navštívil dříve nebo zájezdy, které se podle jeho profilu před-načtou dopředu.

---

<sup>2</sup> <http://m.sunnysoft.cz>

<sup>3</sup> <http://www.tactustechnology.com/technology.html>

### 2.1.5 Výkon

Výkon chytrých telefonů a tabletů dnes roste neskutečným tempem. Je možná až zbytečné v ohledu na prohlížení webu nebo obyčejné prohlížení informací řešit omezení výkonu.

### 2.1.6 Shrnutí

Z několika předešlých odstavců bychom si mohli vyvést následující závěry. Mobilní zařízení má mnohé výhody, ale také problémy a omezení. Pro vývojáře, který chce zákazníkovi nabízet kvalitní informace, vyplývají některá doporučení:

- Je nutné zohlednit velikost displeje.
- Zobrazení informací je dosti omezené ani ne tak rozlišením, ale spíš čitelnou velikostí. Navíc je dobré nezapomenout na špatnou čitelnost na slunci a přizpůsobit tomu barvy.
- V ovládání převládá dotykový displej, velikosti a rozlišení ovládacích prvků musí korespondovat s velikostí ovládacího média – prstů.
- Připojení se sice předpokládá trvalé, ale kvalita a rychlost není vždy na dostatečné úrovni. Je dobré zohlednit velikost stahovaných dat a případně se připravit na možné výpadky připojení.
- Mobilní zařízení je napájeno baterií, při programování je dobré brát ohled na využívání funkcí s vysokou spotřebou energie. Nehledě na to, že nás v tomto ohledu bude omezovat i operační systém.
- Nesmíme zapomenout ani na výkonnostní omezení, zejména pak omezení velikosti operační paměti.

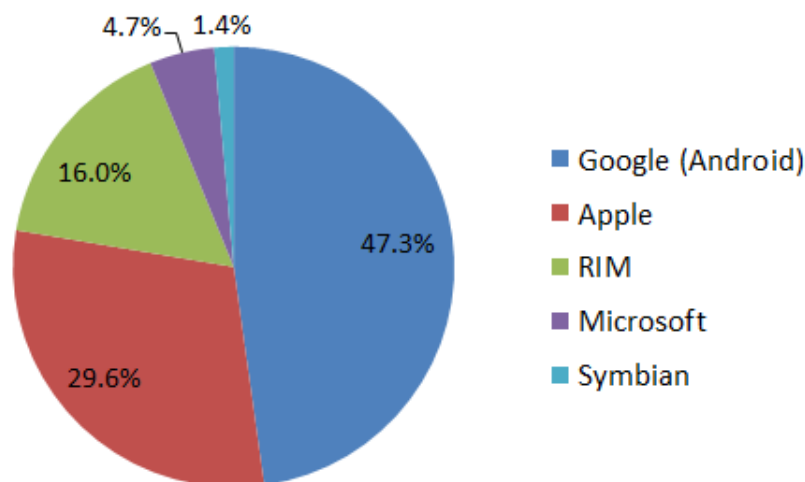
## 2.2 Výběr platformy, výhody, nevýhody, přínosy

Při výběru platformy máme v podstatě tři možnosti implementace – plnohodnotná aplikace, aplikace ve formě webové stránky a hybridní aplikace. Jinou terminologií řečeno jde o těžkého klienta, lehkého klienta a o jejich kombinaci.

### 2.2.1 Plnohodnotná nativní aplikace

Plnohodnotnou nativní aplikací myslíme software přímo instalovatelný a spustitelný na mobilním zařízení. Aktuálně existuje několik významných platforem, kterými jsou mobilní zařízení (chytré telefony) vybaveny. Předpokládám, že do budoucna má momentálně smysl se zabývat OS Android, iOS, Windows Mobile, příp. BlackBerry. Aktuální prodejnosti viz graf na Obr. 1.





Obrázek 1 - Prodejnost podle operačního systému [comScore - únor 2012 [4]]

### Výhody platformem

- Hlavní výhodou je dostupnost API, potažmo služeb, které mobilní zařízení nabízí. Patří mezi ně například přístup k GPS, přístup k datovým komunikacím, informacím o vlastníkovi, přístup k fotoaparátu, přístup k odesílání SMS nebo přímému volání a další.
- Provázání s přístrojem. Instalaci můžeme SW klientovi lépe nabídnout. Nainstalují se spouštěcí ikonky, příp. *gadgets*<sup>4</sup>, které představují uživatelský standard aplikačního uživatelského rozhraní.

### Nevýhody platformem

- Závislost na platformě. Vytvořený sw je závislý na platformě, pro kterou je vytvořen. Dalo by se sice říci, že např. Android a BlackBerry používají pro vývoj Javu, ale každý má své vlastní API a je tedy nutné stejně aplikaci přeprogramovat.
- Závislost na verzi OS. Například u Androidu je verzí za poměrně krátkou dobu velké množství. A každá přináší nějaké změny. Dokonce i v rámci stejné verze na různých přístrojích dochází k přizpůsobením a z toho vyplývajícím odlišným podporám některých funkcionalit. V rámci zamýšlené aplikace, ale pravděpodobně nedojde k nutnosti využívání nestandardních funkcí, které by mohly různé verze ovlivňovat.
- Použití „store“. Aplikace se nabízí přes obchody, které sice usnadňují cestu instalace k zákazníkovi, ale stále vyžadují jeho spolupráci. A nejen to, z pochopitelných bezpečnostních důvodů zákazník musí instalaci také schvalovat. Navíc musí klient aplikaci vyhledat (musí vědět, že v obchodě skutečně je). Dostat aplikaci do obchodu také není zcela triviální [5].

<sup>4</sup> Gadget – miniaplikace instalovatelná do chytrých telefonů. Poskytují částečnou funkcionalitu aplikace a slouží jako „lákadlo“ pro její spuštění viz [31].

## 2.2.2 Aplikace ve formě HTML, CSS s podporou javascriptu

Přehled obecných vlastností tenkého klienta.

### Výhody

- Napříč platformami. Přirozeně jako nejhodnotnější výhodu považují skutečnost, že HTML pracuje na všech platformách s podporou prohlížení webu.
- Nasazení. Jestli se vůbec dá hovořit o nasazení, pak je nejjednodušší, protože nevyžaduje žádnou instalaci. Uživatel nemusí schvalovat pravidla, přemýšlet o volném místě, bát se zneužití dat, stačí jen webovou stránku zobrazit.
- Žádný „store“. Pro rozšíření aplikace jsme oproštěni o store, přes který je aplikace nabízena zákazníkům. Není nutné platit členské poplatky nebo čekat na schválení aplikace ze strany storu (např. u Apple).
- Okamžitá aktualizace. Novou verzi aplikace stačí nahrát na webový server a uživatelé k ní mají okamžitý přístup.

### Nevýhody

- Nebezpečí nekonzistence zobrazení. Různá zařízení s různými prohlížeči mohou stránky zobrazit různě. V krajních případech se stránka rozpadne takovým způsobem, že nebude ani čitelná.
- Závislost na připojení. Zobrazení stránek z webového serveru vyžaduje stálé připojení k internetu. Nepočítám-li aplikace nahané offline (třeba i s podporou HTML5) do zařízení. Zamýšlená aplikace bude vyžadovat trvalé připojení k internetu v každém případě, takže v tomto ohledu nejde o nevýhodu.
- Nevypočitatelný javascript. Nemluvě o tom, že javascript jako takový je častým důvodem programátorova zoufalství, vzniká u mobilních zařízení další problém. Pokud chceme pomocí javascriptu simulovat složitější aplikaci, navíc s většími nároky na paměť, pravděpodobně narazíme na omezení mobilních prohlížečů, nebo alespoň budeme plýtvat pamětí více, než by tomu tak bylo u nativní aplikace, kde je možné vše lépe ohlídat.
- Výkon. Obdobně jako u paměti je jistě možné dosáhnout větší výkonnosti u nativních aplikací.

## 2.2.3 Hybridní aplikace

Hybridní aplikace spojuje výhody a také nevýhody možností uvedených výše. Řekněme, že nativní aplikace je jen jakousi schránkou pro prohlížeč HTML. Android i Windows Mobile má takovýto prohlížeč přímo součástí základního frameworku, takže jednoduchou aplikaci se zobrazením HTML stránky je možné napsat na několika málo řádcích. Obdobným šalamounským přístupem si

práci ulehčil Sunnysoft u své mobilní aplikace<sup>5</sup> pro systém Android. Otázkou ale je, co nám to může přinést.

Na první pohled se jedná o řešení hodné Potěmkinovy vesnice, kdy se navenek tváří jako instalovaná aplikace a uvnitř práci odvede HTML. Může však také pomoci spojit výhody obou řešení. Samotná aplikační schránka může zajišťovat potřebný přístup k API platformy a tím zpřístupnit informace nebo služby webové stránky nedostupné. Případně poskytuje vrstvu pro uchovávání informací v čase. HTML5 sice offline úložiště nabízí, ale aktuálně není HTML5 na mobilních platformách dostatečně rozšířeno. A zobrazovaná webová stránka pak zahrnuje obsahové jádro společné pro všechny platformy. Nemusíme se pak pro každý přístroj zabývat zobrazením informací, i když ani to není zaručeno, jak jsem již uvedl v odstavci Aplikace ve formě HTML.

Pro zajímavost uvedu i open source projekt PhoneGap<sup>6</sup>, který zapouzdří HTML do několika platform. Aktuálně: Android, iOS, Windows Mobile, Symbian, BlackBerry. Zajistíme si tak jednoduše přístup k službám zařízení bez nutnosti programovat aplikaci pro každou platformu zvlášť. Navíc s výhodou, že je PhoneGap již před-schválen u společnosti Apple.

## 2.3 Shrnutí

V kapitole 2 jsem přiblížil některá omezení mobilních zařízení. Jedním z největších problémů je velikost displeje. To dává na vývojáře jisté nároky na hospodaření s ovládacími prvky a místem pro zobrazování informací. Proto bude velice vhodné zabývat se dále technikami, které s množstvím informací i komunikací se zařízením pomohou.

Také jsem se zabýval možností implementace a výběrem platformy. Vzhledem k tomu, že zamýšlená aplikace nemá žádné nároky na API přístroje, není nutné implementovat plnohodnotnou aplikaci. Naplno tak můžeme využít výhod webové aplikace, a to zejména dostupnost na všech platformách a jednoduchost distribuce.

---

<sup>5</sup> [https://play.google.com/store/apps/details?id=cz.sunnysoft.android.eshop&feature=search\\_result](https://play.google.com/store/apps/details?id=cz.sunnysoft.android.eshop&feature=search_result)

<sup>6</sup> <http://phonegap.com/>



### 3 Cestovní ruch – úvod do problematiky

Prototypová aplikace pro vyzkoušení možností doporučování informací pro mobilní zařízení bude úzce cílena na doménu prodeje zájezdů. Této doméně se již mnoho let věnuji a mohu proto vystupovat také v roli doménového experta, který bude aplikaci nastavovat. Nejdříve bych rád vymezil základní pojmy.



Obrázek 2 - Zjednodušený třídní diagram - Základní struktura DB v CK

#### 3.1 Zájezd, termín a služby

Interně v systému používáme terminologii podle úrovní v databázové struktuře (viz obrázek 2). Jádrem jsou následující vrstvy: zájezd – termín – služba, kdy zájezd je zastřešujícím prvkem, skládá se z několika termínů a ty pak ze samotných služeb.

Z reálného světa si můžeme zájezd představit jako jednu stránku papírového katalogu cestovní kanceláře, kde je prezentován jeden hotel. Ten má několik turnusů (termínů), kdy každý z nich může mít jiné vlastnosti, například jiné délky trvání, stravu, dopravu, ale i různé druhy ubytování, třebaže jde stále o jeden hotel. Pro každý turnus jsou pak nastaveny ceníky služeb – služby. Ty nesou primárně informaci o ceně a určení.

##### Cestovní kancelář

Přesná definice dle zákona:

*„Cestovní kancelář je oprávněna podle ustanovení zák. č. 159/99 Sb., § 2, odst. 1 na základě koncese nabízet a prodávat zájezdy.*

*Dále je oprávněna rovněž:*

*nabízet a prodávat jednotlivé služby cestovního ruchu a na základě individuální objednávky prodávat jejich kombinace*

*organizovat kombinace služeb cestovního ruchu a nabízet je a prodávat jiné cestovní kanceláři za účelem jejího dalšího podnikání*

*zprostředkovávat prodej jednotlivých služeb cestovního ruchu pro jinou cestovní kancelář nebo cestovní agenturu, případně pro jiné osoby (dopravce, pořadatele kulturních, společenských a sportovních akcí apod.)*

*zprostředkovávat prodej zájezdu pro jinou cestovní kancelář; cestovní smlouva v těchto případech musí být uzavřena jménem cestovní kanceláře, pro kterou je zájezd zprostředkováván prodávat věci související s cestovním ruchem, zejména vstupenky, mapy, plány, jízdní řády, tištěné průvodce a upomínkové předměty“ [6]*

## **Cestovní agentura**

*„Cestovní agentura je oprávněna podle ustanovení zák. č. 159/99 Sb., § 3, odst. 1 - 3 na základě ohlášení vázané živnosti provozovat činnost v rozsahu uvedeném v § 2 odst. 2 písm. a) až d).*

*Cestovní agentura nesmí zprostředkovávat prodej zájezdu podle § 2 odst. 2 písm. d) pro subjekt, který není cestovní kancelář ve smyslu § 2 odst. 1.*

*V rámci živnosti podle odstavce 1 může cestovní agentura rovněž prodávat věci související s cestovním ruchem, zejména vstupenky, mapy, plány, jízdní řády, tištěné průvodce a upomínkové předměty.“ [7]*

Pro naše účely můžeme jednoduše říci, že cestovní kancelář kompletuje zájezdy z několika služeb a nabízí je k prodeji. Cestovní agentura jen předprodává zájezdy cestovních kanceláří.

Původně naše firma fungovala jen jako CA a postupem času a širším nástupem webových prezentací začala nabízet zájezdy na internetu. Pro neexistenci komerčního agregátora dat jsme začali data cestovních kanceláří sami shromažďovat a vyvíjet celé řešení pro cestovní agentury a kanceláře.

Dnes tedy mimo jiné vyvíjíme software pro cestovní ruch, agregujeme řádově miliony zájezdů a nabízíme hotová řešení stovkám cestovních agentur.

## **3.2 Agregace zájezdů**

Málokdo by věřil, že se nabízení zájezdů a dat o nich nachází v tak dřevní době, jako je tomu u nás v ČR. Bohužel neexistuje obecně přijatá norma nebo univerzální formát dat. Dá se říci, že každá cestovní kancelář má vlastní formát nabízených dat, a to včetně vlastních číselníků. Při zpracovávání dat se bohužel potýkáme i s velkým množstvím chyb a rozdílná terminologie dokonce vyžaduje ruční zařizování množství informací vyškolenými pracovníky. O zbytek se stará importovací robot, který unifikuje informace CK do jednotného formátu a názvosloví. Jeho bližší popis podléhá utajení firmy. Pro potřebu vyhledávání a adaptace máme k dispozici široké portfolio rozříděných a přeložených informací. Všechny informace jsou navíc několikrát denně aktualizovány. Nabízí se jejich širší využití, které je také náplní této práce.

## **3.3 Současný stav a konkurence**

Současný stav přechodu některých internetových prodejců v Česku na mobilní zařízení hodnotí [8]. Článek se však přímo netýká cestovního ruchu, důvodem je totiž neexistence aplikací. Autoři článku vytýkají aplikacím nepřehlednost, nekonzistenci, pomalost, nedomyšlenost funkcí a dokonce i barevný návrh. Kladné ohlasy byly spojovány většinou s přehledností, rozsahem služeb a rychlostí.

Sám jsem si mnohé aplikace otestoval a musím říci, že mne osobně nejvíce zamrzely nedotažené funkcionality, jako nemožnost dokončit objednávku nebo přihlásit se na webový účet.

Ostudná je aplikace od největšího prodejce u nás - mall.cz, která mne vždy jen uvítala hlášením o tom, že nejsem připojen k internetu. Poznamenávám, že popisuji stav, jak jsem jej neformálně testoval v srpnu 2012.

Je zřejmé, že mnohé aplikace trpí dětskými nemocemi, a to nejen aplikačními chybami ale i chybami návrhu. Není se čemu divit, lidé se teprve seznamují s možnostmi nakupování přes chytré telefony a prodejci se jistě budou postupně potřebám zákazníků přizpůsobovat.

Snažil jsem se prohledávat internet a z největších prodejců zájezdů jsem mobilní aplikaci ve formě webové stránky objevil jen u společnosti Invia<sup>7</sup>. Ostatní buď vyčkávají, mají aplikaci ve vývoji nebo se jim ji podařilo (přede mnou) dostatečně utajit.

Invia jde cestou dvoufázového vyhledávače s jednoduchým výpisem podle termínů zájezdů a přehledným detailem zájezdu s poněkud krkolomnou možností objednávky. Co mne docela překvapilo, nikde na stránkách nenaleznete fotografie, což je dle mého názoru, u nabídky zájezdů dost zásadní problém. Informace jsou sice zobrazeny přehledně, ale termínový výpis a nepřítomnost fotografií<sup>8</sup> jej zcela devalvují. Jako vyhledávač letenek by dostačoval, ale jako varianta cestování "prstem po mapě" ve chvílích čekání na zastávce rozhodně nedostačuje.

### **3.4 Nabídka zájezdů – [www.zajezyonline.cz](http://www.zajezyonline.cz)**

Firma, ve které pracuji, se mimo tvorbu SW pro cestovní ruch, zabývá také prodejem zájezdů jako cestovní agentura. Proto máme rozsáhlé praktické zkušenosti s nabízením zájezdů přes internet. Pokud se bavíme o možnosti nabízení a doporučování zájezdů na mobilním zařízení, rád bych nejdříve popsal, jaká data a jaké stránky nabídkový web zahrnuje. Konkrétně budu pracovat s webem [zajezyonline.cz](http://zajezyonline.cz)<sup>9</sup>. Všechny níže uvedené sekce se budeme snažit dostat do jejich mobilních mutací.

#### **3.4.1 Hlavní strana**

Hlavní strana je seskládána z množství panelů, které postupně popíši. Základním panelem je vyhledávání. Poté jde o několik panelů se seznamy zájezdů, jako jsou např. nabídky last minute. Zbytek stránky ponechám stranou, protože pro mobilní verzi nejsou až tak důležité a pro úsporu místa je eliminujeme - například články o cestování, novinky atp.

---

<sup>7</sup> <http://www.invia.cz>

<sup>8</sup> Těsně před dokončením diplomové práce Invia začala fotografie v omezené míře zobrazovat.

<sup>9</sup> <http://www.zajezyonline.cz>

### 3.4.2 Vyhledávač zájezdů

The screenshot shows a travel search interface with a yellow background. At the top, there are four tabs: 'Všechny zájezdy' (selected), 'Jen Last Minute', 'Jen First Minute', 'Číslo zájezdu', and 'Poptávka'. Below the tabs, there are several search criteria: 'Země: Bulharsko' with a plus icon, 'Destinace: Albena' with a plus icon, 'Odjezd od:' with a calendar icon, 'Návrat do:' with a calendar icon and 'délka: -', 'Doprava: Letecká', 'Stravování: Minimálně Vlastní', 'Letiště: Praha' with a plus icon, 'Ubytování: Hotel', 'Zaměření: nerozhoduje', 'Úroveň hotelu: nerozhoduje', 'Cena: chci zadat přesně ->', 'Hledání v textu:', 'Cena od: Do: Kč', and 'Vybavení: Vyberte si vybavení' with a plus icon. At the bottom left, it says 'Počet aktuálních termínů splňujících kritéria: 315'. At the bottom right, there is a button labeled 'Zobraz zájezdy'.

Obrázek 3 - Formulář hledání zájezdů

Jak je vidět na náhledu, vyhledávací formulář je dosti obsáhlý. Kritéria by se dala rozdělit do těch více používaných a do těch, která jsou používána méně.

Kritéria vyhledávání můžeme dělit několika způsoby:

Podle použití:

- více používané – například země, destinace, termín
- méně používané

Podle viditelnosti:

- viditelné hned
- viditelné v závislosti na výběru - například výběr letiště se objeví až po zvolení letecké dopravy
- neviditelné – existuje i mnoho parametrů klientovi naprosto nedostupných, sloužících pro systémový výběr

Podle množstevnosti:

- jedno-výběrové
- více-výběrové – například destinací můžete zadat více najednou

Záměrně jsem nepoužil dělení podle důležitosti. Pro každého uživatele může být důležité něco jiného a předpokládám, že další analýzou se nám podaří zjistit, která kritéria, případně jejich podskupiny, jsou pro které klienty zajímavé. Věřím, že segmentace zákazníků by mohla vést k přehledné nabídce kritérií podle preferencí.


Z náhledu je také jasné, že právě vyhledávání je největší problém vzhledem k malému displeji mobilního zařízení. Není možné všechna kritéria vypsat ve viditelné části obrazovky. Dále se tedy budu také zabývat možnostmi zjednodušení komunikace, zavedení dialogu a usnadnění zadávání preferencí.

### 3.4.3 Nabídka výsledků vyhledávání

Podle zadání kritérií ve vyhledávači, se vypisuje seznam odpovídajících výsledků vyhledávání. Jde o seznam zpravidla deseti zájezdů, doplněných o 3 termíny pro každý zájezd.


HOTEL KOMPAS | Hotel ★★

od **11 790 Kč**  
(28.6. (Čt) -1.7.2012 (Ne))



Bulharsko, Albena

TERMÍN	DNY	DOPRAVA (odkud)	STRAVA	CENA
28.6. (Čt) -1.7.2012 (Ne)	4	✈ Praha	All Inclusive	11 790 Kč + -
30.8. (Čt) -2.9.2012 (Ne)	4	✈ Praha	All Inclusive	11 790 Kč + -
9.8. (Čt) -12.8.2012 (Ne)	4	✈ Praha	All Inclusive	11 990 Kč + -


POSLAT

ZOBRAZIT VŠECH 43 TERMÍNŮ, KTERÉ SPLŇUJÍ VAŠE POŽADAVKY

ZOBRAZIT DETAIL ZÁJEZDU

Obrázek 4 - Položka výsledku hledání


Takovýto způsob výpisu bude muset být také revidován, protože je pro chytrý telefon až příliš obsáhlý. Cílem je zobrazit jen důležité informace a hlavně pokud možno zobrazit jen ty nejvhodnější nabídky, tak aby bylo rolování na stránce a také stránkování omezeno na minimum. K dalšímu zkoumání nechávám právě zajištění výpisu nejvhodnějších nabídek.

### 3.4.4 Detail zájezdu

Detail zájezdu slouží k zobrazení všech potřebných informací o zájezdu. Tato strana by se dala rozdělit do několika sekcí.



## HOTEL KOMPAS To se mi líbí

 Bulharsko - Albena

Nepodařilo se ověřit volné kapacity, zájezd by však měl být k dispozici.

**Proveďte nezávaznou rezervaci.**

Termín: (čt) 28.06.12 - (ne) 01.07.12

[ZVOLIT JINÝ TERMÍN](#)

[VOLNO: REZERVUJTE ZÁJEZD!](#)

Počet dní: 4 dny

Ubytování: Hotel ★★☆☆

Doprava: Letecká z: Praha

Stravování: All Inclusive


Zaměření: Pobytové


Cena od: 11 790 Kč



 [POSLAT NA EMAIL](#)

 [VLOŽIT DO SCHRÁNKY](#)

 [ZOBRAZIT NA MAPĚ](#)

 [SDÍLET NA FACEBOOKU](#)

Číslo zájezdu: 35926 - 13818201

**POPIS ZÁJEZDU**

[TERMÍNY](#)

[FOTOGALERIE](#)

[CENY A NEZÁVAZNÁ REZERVACE](#)

 [DISKUZE](#)

 [MAPA](#)

SLEVY PRO TENTO ZÁJEZD:



### POPIS ZÁJEZDU

#### Poloha

Hotel se nachází v severní části střediska Albena v krásné přírodě, obklopený zelení ve vzdálenosti cca 400 m od rušného centra.

#### Pláž

Krásná písčná pláž je 600 m. Dopravu na pláž zajišťuje hotelový autobus

### DOKLADY A VÍZA

Při cestách do zahraničí si nezapomeňte zkontrolovat platnost Vašeho pasu či dalších dokladů potřebných ke vstupu do této země!

Více informací o potřebných

Obrázek 5 - Náhled GUI aktuálního systému - Detail zájezdu

- informace o zájezdu a termínu
- nabídka dalších termínů zájezdu
- popis
- fotografie
- nabídka služeb
- objednávka
- diskuze

Rozdělení uvádím také proto, že by mohlo být vodítkem pro selektivní zobrazení detailu zájezdu na malém displeji. Z náhledu je jasné, že nemá smysl snažit se zobrazit všechny informace najednou, ale ve formě adaptivní prezentace zobrazení relevantních informací s využitím skrývání jednotlivých sekcí.

## 3.5 Shrnutí

Kapitola 3 byla věnována doméně cestovního ruchu, konkrétně nabízení zájezdů v prostředí internetu. Byly vysvětleny základní pojmy a byla představena stávající webová aplikace cílená na klasická PC. Podrobněji jsem rozepsal jednotlivé komponenty stránek, které bude nutné upravit pro mobilní platformu. Největším problémem je množství informací, nabídek a rozsáhlost vyhledávacího formuláře. Také jsem nastínil nutnost hledání vhodné nabídky, pokud možno co nejvíce vyhovující preferencím uživatele. Všechny tyto oblasti se budu snažit v dalších kapitolách řešit pomocí doporučování informací.

## 4 Doporučování informací

Stále více zákazníků utrácí své peníze raději na internetu než v kamenném obchodě. Eshopy nabízí jednoduchý nákup z pohodlí domova, jsou schopny představit mnohem širší nabídku a obsloužit zákazníka bez ohledu na jeho lokaci. Velikost nabídky ale spolu nese riziko, že zákazník nebude schopen relevantně rozhodnout, který produkt je pro něj vhodný, nebo dokonce nebude schopen produkt vůbec najít. Komfort hledání je snížen množstvím kategorií, vlastností výrobků a složitých formulářů pro hledání. Pro zákazníka neznalého problémové domény to může vést k frustraci a následnému opuštění obchodu.

Na mobilním zařízení je situace ještě horší. Prostor pro zobrazení vyhledávacího formuláře a všech jeho parametrů je omezen. Stejně tak je omezen prostor pro zobrazení nalezených produktů. Orientace v kategoriích a možnosti vstupu jsou také horší. Pohodlí výběru metodou single-shot, popsané níže, klesá a je vhodné se zamyslet nad vhodnou alternativou. Tou může být systém doporučování.

Systémy doporučování mohou pomoci zákazníkovi s výběrem. Mohou zjednodušit komunikaci, omezit množství požadovaných informací, či snížit počet iterací hledání. S nadsázkou se dá říci, že takový systém simuluje prodejce, který komunikuje se zákazníkem. Postupně se jej doptává na požadované vlastnosti a nakonec je schopen doporučit produkt pro každého jednotlivého zákazníka. To pomáhá zvýšit loajalitu a spokojenost klienta, který se bude rád vracet. Překvapivě je v dnešní době takový systém stále konkurenční výhodou, protože jím mnoho eshopů nedisponuje. Tím méně v oblasti mobilních zařízení. Primárně v rámci menších eshopů se může stát klíčovou konkurenční výhodou.

### 4.1 Základní dělení

Metod doporučování existuje celá řada. Stejně tak jsem v literatuře našel několik možností jejich dělení (např. [9]). Nejzákladnější rozdělení je podle orientace na uživatele – personalizované a nepersonalizované.

Jednoduše lze říci, že u nepersonalizovaných metod je doporučení vydáváno bez ohledu na uživatele a za stejných okolností je nabídnuto všem stejné doporučení. Nejzákladnější metodou je Filter-Based Recommendation – FBR, doporučení na základě přesné shody. Uživatel zadá dotaz a na jeho základě je vypsáno několik nabídek, obdoba SQL dotazu na databázi. Otázkou je, zda se v takovém případě dá vůbec mluvit o doporučení. Mezi další nepersonalizované metody patří například výpis několika top-N produktů, které jsou:

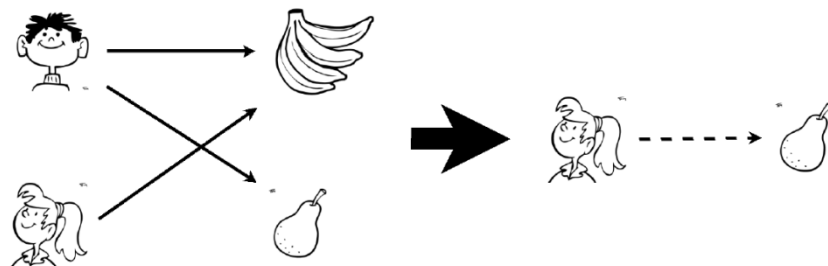
- nejvíce kupované,
- nejvíce prohlížené,
- nejlépe hodnocené,
- atp.

Dále se budu zabývat personalizovanými metodami. Nejčastěji se používá dělení na tři hlavní proudy – kolaborativní, založené na znalostech a založené na obsahu. Mimo tyto kategorie existují také další, například demografické, ty ale nebudou dále rozebírány.

Nyní uvedu popis, dělení, výhody a nevýhody základních proudů personalizovaných metod.

#### 4.1.1 Kolaborativní metody - Collaborative based

Základní myšlenkou je předpoklad, že podobní uživatelé budou preferovat podobné produkty. Využívá se uživatelská historie, zpravidla ve formě hodnocení. Doporučení vychází z profilů uživatelů podobných tomu aktivnímu. Budou doporučeny produkty, které uživatel ještě nemá ve svém profilu, ale jsou vysoce ceněny jím podobnými.



Obrázek 6 - Kolaborativní metody [Universal recommender system [10]]

Výhodou je doménová nezávislost. Nepotřebujeme mít tedy vysokou znalost o produktech – jejich specifikaci ani popisu.

Nevýhodou kolaborativního přístupu je závislost na uživatelském hodnocení. Pokud je velké množství položek bez hodnocení, nebo je aktuální uživatel nový a tedy bez známých preferencí, je doporučení špatně dostupné. Tomuto problému se říká studený start.

Podle přístupu, jakým jsou předpovídaný nové vazby, dělíme kolaborativní metody do dvou skupin: memory based a model based.

##### Memory based

Využívá se všech nebo části uživatelských hodnocení pro generování předpovědi. Každý uživatel je součástí skupiny se stejnými zájmy. Algoritmus založený na hledání okolí pracuje se skupinou uživatelů, vybranou na základě podobnosti s aktivním uživatelem. Z jejich hodnocení jsou vypočítány váhy a ty jsou pak výchozí pro získání předpovědi. V knize [11] popisují proces ve třech bodech:

1. Přiřadit váhu všem uživatelům s ohledem na podobnost s aktivním uživatelem.
2. Vybrat  $k$  uživatelů, kteří jsou nejpodobnější s aktivním uživatelem.
3. Vypočítat předpověď z okolních hodnocení.

Blíže pak v [11]. Nevýhodou jsou mimo jiné vysoké nároky na CPU a paměť. Je proto vhodná pro relativně malý počet uživatelů a produktů.

##### Model based

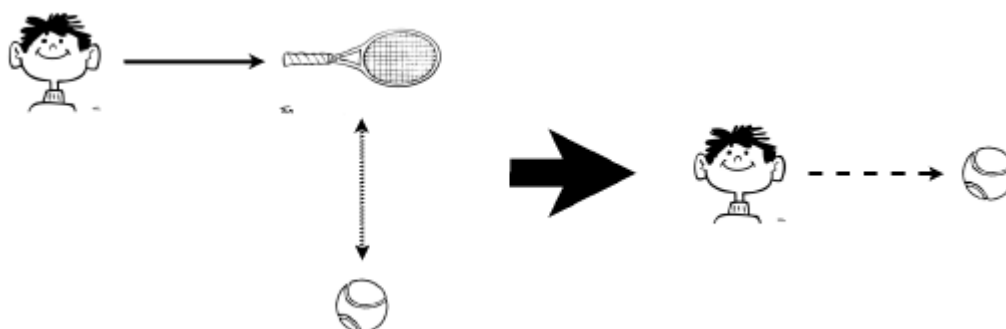
Důležitým znakem této metody je model. Ten slouží k předpovídaní vztahů a získávání doporučení. Model vychází z již známých vztahů mezi uživateli a položkami. Informace jsou ukládány

do vektorů a to umožňuje jejich projekci do multidimenzionálních prostorů. V takovém prostoru jsme schopni zachytit vztahy uživatelů a v jejich blízkosti také položky. Předpoklad je takový, že položky v blízkosti podobných uživatelů, budou také oblíbené novým uživatelem, a proto mu budou doporučeny. Systém se učí položky sám pomocí různých technik z již známých dat.

Jednou z výhod je nízká náročnost na CPU i paměť.

#### 4.1.2 Metody založené na obsahu - Content based

Základním principem této metody je doporučení takových položek, které jsou podobné položkám vybraným uživatelem v minulosti. Na obrázku 7 je patrné, že uživatel má vztah k tenisové raketě. Raketa je podobná na další položku – míček. Můžeme předpokládat budoucí vazbu mezi uživatelem a míčkem. Proto mu můžeme míček doporučit.



Obrázek 7 – Metody založené na obsahu [Universal recommender system [10]]

Využívá se primárně u produktů založených na textu, například web nebo zprávy. Tedy u produktů, pro které je možné najít textová interpretace. Produkty jsou pak popsány pomocí asociovaných vlastností - např. klíčových slov.

Nevýhodou této metody je především nutnost zadání velkého množství informací o položkách a jejich vazbách. Také je vyžadována jistá znalost o uživateli.

#### 4.1.3 Metody založené na znalostech - Knowledge based

Jak jsem uvedl výše, kolaborativní i content-based systémy jsou vhodné pro produkty jako zprávy, také produkty ohodnotitelné kvalitativně jako knihy nebo filmy. Ale například pro produkty, které nekupujeme tak často, tudíž je těžké nasbírat uživatelská hodnocení, se nehodí. Pro takové jsou řešením metody založené na znalostech [12].

Výhodou systému založeného na znalostech je jeho nezávislost na množství uživatelského hodnocení, jako tomu je u kolaborativních metod. Jsou také lépe odolné vůči změnám preferencí uživatele.

Existují dva základní přístupy

- Case based – založeno na podobnostních mírách nebo metrikách
- Constraint based – využívá sadu definovaných pravidel a omezení

## Constraint based

Je založen na předdefinovaných znalostních bázích, které obsahují jasná pravidla o tom, jak souvisí požadavky s produkty. Uživatel zadá své preference a na základě pravidel jsou položky doporučeny - funguje tedy obdobně jako klasický vyhledávač.

Nevýhodou je, že takové hledání není personalizované a vyžaduje nastavení pravidel ze strany doménového experta. Tato pravidla mohou být velmi komplikovaná, pokud chceme dosáhnout dobrého doporučení s ohledem na kontext.

## Case-based

Protože se budu touto metodou blíže zabývat v kapitole 4.2, uvedu jen základ potřebný k rozhodnutí.

Na začátku se systém chová jako FBR, kdy na základě požadavku vyhledá v katalogu odpovídající položky. Jakmile ale uživatel projeví o položku zájem, například ji objedná, položka se společně s požadavkem uloží v podobě případu do case-base. Tyto případy jsou poté použity pro doporučování v budoucnu. Pokud pak uživatel zadá nový požadavek, ten je porovnán s požadavky z minulosti, nejpodobnější je vybrán a podle uloženého řešení jsou vybrány položky k doporučení pro aktuální hledání.

Metoda je vhodná pro položky s dobře definovaným souborem vlastností. Nemá nevýhody jako je studený start a umí se dobře vypořádat se změnami nálad uživatele – s vysokým ohledem na kontext. Doporučení nemusí být stoprocentní, protože ani uživatel v minulosti nemusel vybrat dokonalé řešení.

### 4.1.4 Výběr metody

Pro přehlednost uvedu základní směry s jejich výhodami a nevýhodami (viz tabulka 1).

Metoda	Výhody	Nevýhody
Kolaborativní	Doménová nezávislost Není nutné zjišťovat informace o položkách	Studený start Vyžaduje velké množství dat hodnocení uživatelů
Založené na obsahu	Doménová nezávislost	Studený start Vyžaduje zadání vztahů a vlastností
Založené na znalosti	Netrpí studeným startem Nevyžaduje explicitní hodnocení ani definici vztahů	Uživatel musí zadat preference

*Tabulka 1 - Přehled metod doporučování*

Zamýšlený projekt nebude mít kvůli velmi objemnému katalogu pro každou položku dostatek hodnocení, kolaborativní metody jsem tedy rovnou zamítl. Také metody založené na obsahu mají poměrně komplikovanou definici vztahů s nutností mít dostatek informací o uživateli z minulosti. Opět jsou pro velké množství dat téměř nepoužitelné, a to i z důvodu vysokých nároků na výpočetní prostředky.



Jako velice vhodným řešením se tak nabízí metoda doporučování založená na case-based reasoningu. Zejména proto, že netrpí studeným startem, nevyžaduje množství hodnocení pro každou z položek a na jejím základě jsme schopni doporučit i v případě, že o uživateli nemáme dostatek informací z minulosti.

## 4.2 Case-based doporučení

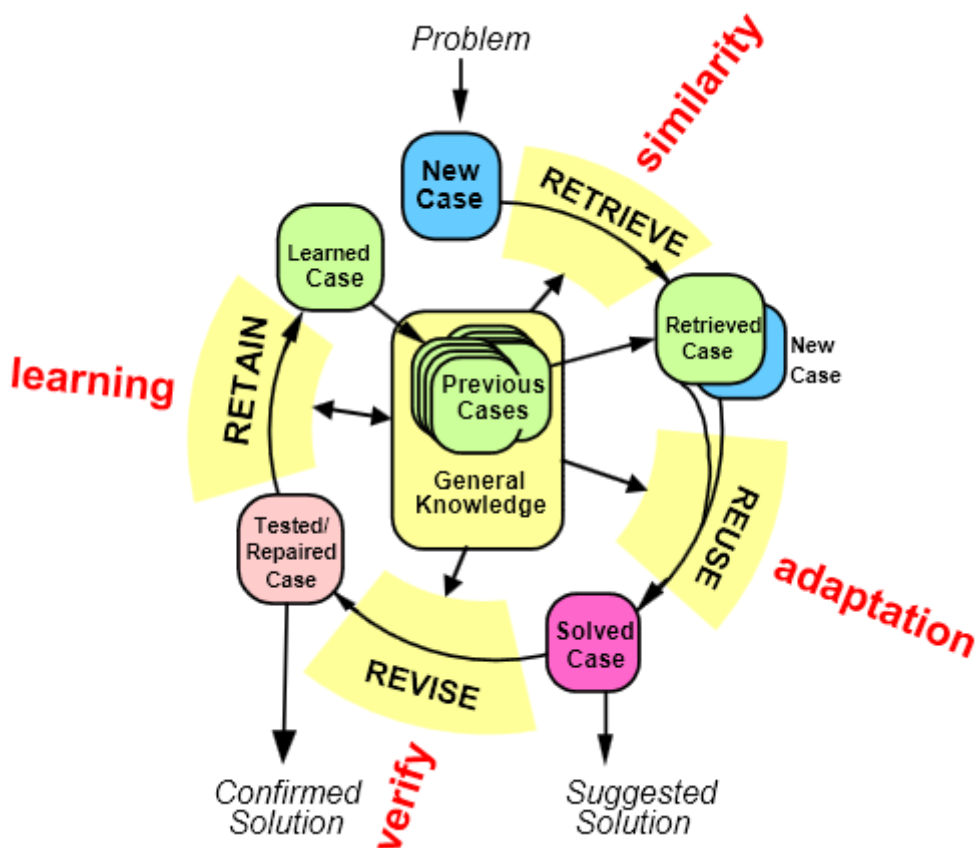
Některé zdroje tuto metodu řadí do content-based jiné do knowledge-based. Do jisté míry využívá také výhod kolaborativních metod. Protože je tato forma doporučení postavena na rozhodování založeného na případech (CBR), nejdříve popíšeme právě tento algoritmus.

### 4.2.1 Rozhodování založené na případech - Case Based Reasoning

Case base reasoning (dále jen CBR) je založeno na myšlence popsané touto citací:

*Rozhodování založené na případech řeší nové problémy adaptací řešení, které bylo použito na řešení starých problémů. – [13].*

Obvykle je proces rozhodování popisován na čtyřech fázích podle schématu na obrázku 8.



Obrázek 8 – CBR [viz [14] podle [15]]

Nyní již detailně rozepíši každou fázi.

**Načtení (Retrieve)** – Získání případů, které nejvíce odpovídají aktuálnímu problému (*New case*), z báze případů (*case-base*), je v obrázku označeno jako *Previous cases*. Nejdůležitější komponentou je určování podobnosti. Na základě nastavených měr a vah jsou vypočítány úrovně podobnosti a podle nich případy seřazeny. Pro další zpracování je vrácen jeden nebo více nejpodobnějších případů - *Retrieved cases*, označované také jako cílové (*target-cases*).

**Opětovné použití (Reuse)** – Řešení, která byla použita u případů v minulosti, se adaptují na nový problém.

**Revidování (Revise)** – Získané řešení je aplikováno a z něj získáme zpětnou vazbu, například zda bylo nebo nebylo úspěšné, nebo jak bylo nutné postup upravit.

**Uložení (Retain)** – Pokud bylo řešení úspěšné, je uloženo do báze případů i s použitým postupem.

Názorný příklad popisují v [14]. Komentuje nalezení řešení, jak opravit auto, kdy na vstupu je několik vlastností popisujících závadu a pomocí předchozích případů je doporučeno řešení pro aktuální problém.

Podobnost přináší výhodu oproti technikám přesné shody, jako je například konvenční databáze nebo metody *constraint based*. Pokud je například uživatelův dotaz přespecifikován (*over-specified*), nevrátí tyto metody žádný výsledek, proti tomu CBR najde položky podobné. Nebo naopak, pokud je dotaz pod-specifikován (*under-specified*) a systém najde velké množství výsledků, CBR je schopno vyřadit nejméně podobné položky.

CBR systémy jsou primárně závislé na dvou komponentách: posuzování podobnosti a reprezentace.

## 4.2.2 Posuzování podobnosti

Pro CBR je klíčové posuzování podobnosti, a to z několika důvodů. Poprvé je využito při hledání nejvhodnějšího případu na základě podobnosti problému. Také ve fázi adaptace na aktuální problém můžeme míru podobnosti využít při srovnávání kandidátských položek s tou uloženou v případě. V obou fázích je výsledek doporučení závislý na kvalitě posouzení podobnosti.

Hledáme-li nejpodobnější případ, hledáme vlastně podobnost mezi problémy - požadavky. Ty jsou složeny z vlastností a jejich hodnot. Výpočet podobnosti začíná u vlastností, které jsou sdíleny oběma porovnávanými problémy. Každá vlastnost má definovanou funkci pro výpočet podobnosti. Pro všechny páry hodnot jsou tak získány dílčí skóre udávající míru shody.

Abychom získali celkovou podobnost případů, je nastavena funkce sloučení všech dílčích skóre. Ve většině případů se využívá ještě vah vlastností, které zjemňují výpočet a upřesňují výsledek.

$$\text{sim}(t, c) = \frac{\sum_{i=1..n} w_i \times \text{sim}(t_i, c_i)}{\sum_{i=1..n} w_i}$$

Rovnice relativní podobnosti, kde  $t$  je požadavek,  $c$  je případ,  $w$  je váha a  $n$  je počet vlastností [16].

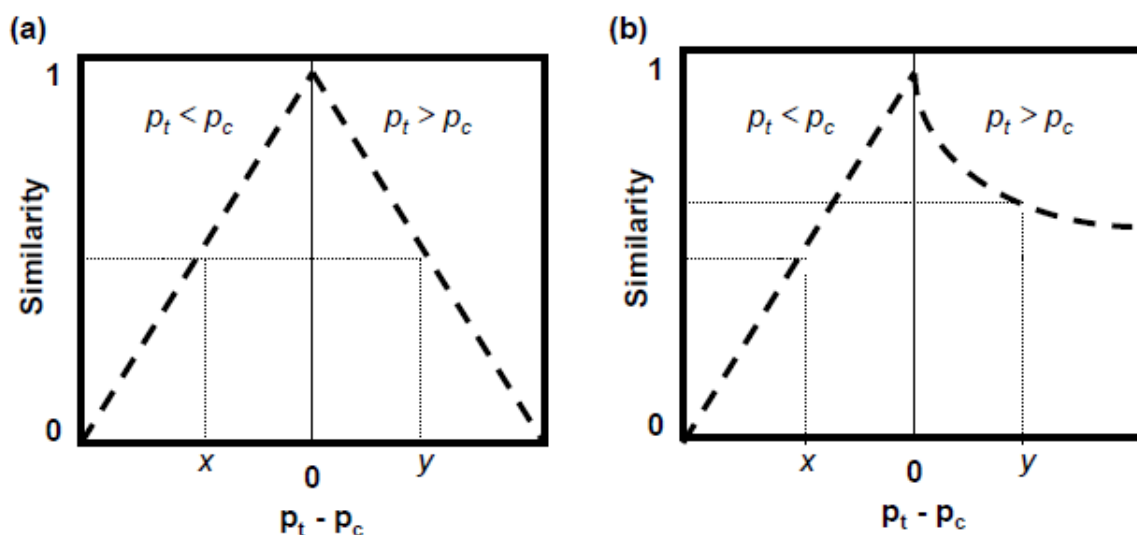
Definice funkcí pro dílčí i celkový výpočet je již prací doménového experta. Jsou posuzovány vlastnosti podle typu, a to nejen číselné, ale třeba i taxonomické stromy. Ty již vyžadují větší znalost

domény a mohou obsahovat i mikrováhy nastavené přímo u částí stromu. Blíže se tímto tématem budu zabývat již konkrétně v kapitole 5.4.

Váhy podobnosti mohou výrazně ovlivnit výsledek srovnání. Pokud například vypočítáme vysokou míru podobnosti hodnot nějaké vlastnosti, můžeme výsledné skóre významně devalvovat nízkou vahou. Proto jsou využívány jako důležitý prostředek pro zjemnění výpočtu. V rámci různých kontextů mohou být váhy nastaveny různě a pro stejné funkce pak budou různé i výsledky srovnání. Definice vah je také v rukou doménového experta. Pro jejich korekci se otevírá prostor pro učení.

## Nesymetrické podobnosti

Ve většině případů se používá podobnosti symetrické. To ale není vždy vhodné. Vezměme si příklad z [17] s cenou. Na obrázku 9 a) je znázorněna symetrická podobnost, kde pro dva různé produkty ( $x$  a  $y$ ), ale stejně vzdálené od středu (např. ceny 500 a 1500) je podobnost identická. V reálném životě je ale třeba zrovna cenu vhodnější řešit nesymetrickou podobností – varianta b). Zde jsou opět produkty  $x$  a  $y$ , opět jsou stejně vzdálené od středu, ale jejich podobnost se už liší. Jednoduše řečeno, produkt s nižší cenou je „podobnější“ než ten s cenou vyšší.



Obrázek 9 - Nesymetrická podobnost [Case based Recommendation - Barry Smith], (Vodorovná osa  $p_t - p_c$  označuje rozdíl cen.)

### 4.2.3 Reprezentace CBR

Reprezentací CBR jsou myšleny postupy a možnosti popisu případu a měř podobnosti. Důvodem je mimo jiné snaha o zdokumentování případného rozhraní pro další využití případů. Informace nashromážděné CBR systémem nemusí sloužit jen pro tuto část systému, ale je dost pravděpodobné že budou využity nebo sdíleny s dalšími částmi projektu. Například v e-komerci se nabízí napojení na CRM a podobně. Míry podobnosti dokumentujeme také proto, aby je byl schopen nastavit člověk neznalý kódu, případně pro pozdější rozšíření v podobě GUI implementovaném na platformě nezávislé na CBR vrstvě. Nesmíme také zapomenout na případná výkonnostní omezení, která pohodlnost a přehlednost popisu může přinést.

V dokumentu [18] jsou popsány základní metody reprezentací. V průběhu času byly také snahy o standardizaci - například CASUEL, CARET, CBML a další. CARET a CBML jsou založeny na XML. Tento formát přináší přehlednost, rozšiřitelnost a nezávislost na platformě. Právě kvůli těmto kladům se budu dále zabývat popisem pomocí XML, konkrétně CBML.

Vývoj CBML započal někdy v roce 1998, kdy jej popsali Hayes a Cunningham. Prošel několika verzemi a vylepšeními. Ve verzi 3 obsahuje strukturální a typovou validaci pomocí XML-schématu tedy CBML – schématu. Odděluje dva dokumenty:

1. *case content document* – obsahuje případ
2. *case structure document* – obsahující vnitřní formát případu

Vnitřní formát případ je dán jednotlivými vlastnostmi, které jsou dvojího druhu – komplexní nebo základní. Komplexní zastřešují další komplexní nebo základní vlastnosti a také umožňují hierarchické struktury - například taxonomie. Definice vlastnosti se skládá z párů – název vlastnosti / typ vlastnosti. Vlastnosti mohou také obsahovat omezení, například obor hodnot nebo hranice přípustného intervalu. Popis CBML v doméně prodeje letenek, která je podobná mnou řešené doméně, je k dispozici zde [19].

CBR metoda vyžaduje počáteční zadání preferencí uživatele, na jehož základě je sestaven dotaz pro algoritmus doporučení. Jako jedna z nevýhod CBR je uváděna právě nutnost počátečního zadání, která může uživatele až odradit od užívání systému. Následující kapitola se věnuje možnostem získávání uživatelských preferencí co nejpríjemnější formou.

### 4.3 Single-shot problém vs. konverzační metody

Většina systémů vyhledávání využívá jednorázové (single-shot) doporučení. Uživatel zadá parametry preferencí do formuláře. Na jeho základě se porovnávají produkty v databázi a ty, které přesně odpovídají, jsou vráceny. Pokud neodpovídají představám uživatele, celý proces začíná znovu. Dá se říci, že taková relace má mizivou dobu životnosti. Reálně ale uživatel málokdy ví všechny požadavky hned na začátku a své preference upravuje v průběhu hledání a výběru.

Některé formuláře hledání pro single shot doporučení bývají dosti obsáhlé, kdy uživatel musí vyplnit mnoho otázek, v danou chvíli třeba i irelevantních. Mnohé vyžadují vysokou znalost domény. Takový formulář je pro uživatele těžce použitelný či dokonce až odrazující. V reálném světě doporučení takto nefunguje, cílem je získat preference v příhodnou chvíli, pokud možno relevantní, uživatelsky čitelné, zodpověditelné a v co nejmenším počtu. S tématem se pojí také termíny:

**Over-specification (přespecifikování)** - Přespecifikování je situace, kdy uživatel zadá parametry hledání tak, že již nejsme schopni nalézt relevantní odpověď. Například pokud vyplní příliš mnoho parametrů, nebo pokud se parametry navzájem vylučují. Uživatel musí slevit na svých požadavcích a parametry hledání vyloučit.

**Under-specification (nedostatečná specifikace)** – Nedostatečná specifikace vede k nalezení velkého množství záznamů, došlo tedy k nedostatečnému omezení. Stává se to ve chvíli, kdy uživatel zadá jen málo parametrů, nebo jsou parametry málo omezující. Výsledek, kdy je uživateli vrácena např. polovina báze, jistě není přínosem a je nutno uživatele vyzvat, aby hledání dále specifikoval.

**Query relaxation** - Neboli uvolňování vlastnosti požadavku můžeme chápat jako omezení jeho vlivu na výsledný dotaz. Abychom toho dosáhli, můžeme buď vlastnost úplně zahodit, nebo upravit její hodnotu. Například u číselné vlastnosti můžeme rozšířit dotazovaný interval, nebo pro vlastnosti taxonomického stromu můžeme využít tzv. query-augmentation (viz níže). Protože query-relaxation se využívá zejména v případech přespecifikovaného požadavku, po uvolnění následuje ověření vlivu a případně se postup opakuje, dokud není set zvládnutelný. O omezení dotazu je vhodné uživatele vhodně informovat.

Míra uvolnění je řízena pravidly nastavenými doménovým expertem, případně může být zpřesněna profilem klienta – dlouhodobými nebo krátkodobými preferencemi resp. váhami. V takovém případě jsou uvolněny nejdříve vlastnosti s nejmenší vahou – nejmenším významem pro uživatele.

**Query augmentation** - Query-augmentation slouží mimo jiné k uvolnění požadavku. Jeho aplikací můžeme upravovat hodnotu vlastnosti typu taxonomie změnou úrovně v rámci stromu.

Pro názornost uvedu příklad z [20] v TripleHop. Pokud uživatel zadá „Thai“, systém doplní „Thai Food“. Pokud nenalezne žádný výsledek, přetypuje na „Asian food“ (výše v hierarchii). Pokud tentokrát zadání odpovídá mnoha výsledkům, můžeme zjemnit přetypováním na „Chinese food“ (níže v hierarchii), jež hledal v minulosti.

Úpravy dotazu by měly být doprovázeny komentářem, že k takové situaci došlo. Pokud například v query augmentation poskočíme o několik úrovní v taxonomii výše a z původně vybrané vesničky se jako destinace označí kontinent, je to dost zásadní změna, o které by měl uživatel vědět.

Výše uvedené problémy pomáhá řešit dialog. Systém postupně zaznamenává chování zákazníka (především jeho zpětnou vazbu) a v každé další iteraci je schopen doporučit lépe. Případně postupně získává informace o preferencích tak, aby množství nalezených položek bylo optimální, nedošlo k přespecifikování nebo nedostatečné specifikaci a zároveň, aby uživatel byl pokud možno schopen na otázky odpovědět v pravou chvíli. Dialog je možné řešit pomocí předdefinovaných scénářů nebo dynamicky. Druhá možnost je jistě zajímavější, protože umožňuje pružněji reagovat na situaci a zákazníkova přání.

Pro dynamický iterační způsob konverzace se využívají především dvě metody, které jsou popsány níže:

- metoda dotazování,
- metoda průběžných doporučení.

#### 4.3.1 Metoda dotazování

Jak jsem již uvedl dříve, metoda dotazování je iterační – inkrementální. To znamená, že nevyžadujeme všechny parametry preferencí hned na začátku, ale preference jsou budovány postupným dotazováním. Otázky musí být jednoduché a srozumitelné pro uživatele. Je nutné předpokládat, že uživatel nemá velkou znalost problémové domény. Pro použití na mobilním zařízení nejsou vhodné otázky očekávající odpovědi v prostém textu. Komplikovalo by to zadání uživatelem.



Postup je následující. Pro inicializaci je použito několik lehce získatelných otázek. Podle odpovědí jsou dále ze sady možných otázek vybrány další. Takto postupujeme, dokud není výsledný počet kandidátských položek optimální - zvládnutelný. Některé metody pro výběr otázek jsou uvedeny níže:

**Entropie** – jednoduše řečeno míra neuspořádanosti. Vychází z termodynamiky, ale vhodně popisuje i informace. Vysvětlím na jednoduchém příkladu, který jsem našel zde [21]. Knihovna, ve které jsou knihy pěkně uspořádané v jednotlivých regálech podle autorů a typu literatury, je nízko-entropická knihovna. V knihovně bez systému, kde nemohu najít žádanou publikaci, strávím spoustu času a ohrožuji tím navíc svou nervovou soustavu. Takovou knihovnu lze klasifikovat jako vysoce-entropickou.

Cílem je dosáhnout maximální redukce otázek, proto je jejich výběr na základě entropie určen velikostí informačního zisku každé z nich. Výši zisku vypočítáme rozdílem entropií před rozdělením setu parametrem a po něm. Čím nižší je entropie po rozdělení, tím vyšší je informační zisk. Snažíme se vlastně předpovědět, jak bude uspořádaná báze po zadání kterého parametru.

Atribut **k omezení (constrain)** je vybrán stanovením atributu s nejnižší entropií (nejvyšším informačním ziskem) mezi atributy, které ještě uživatel nevybral.

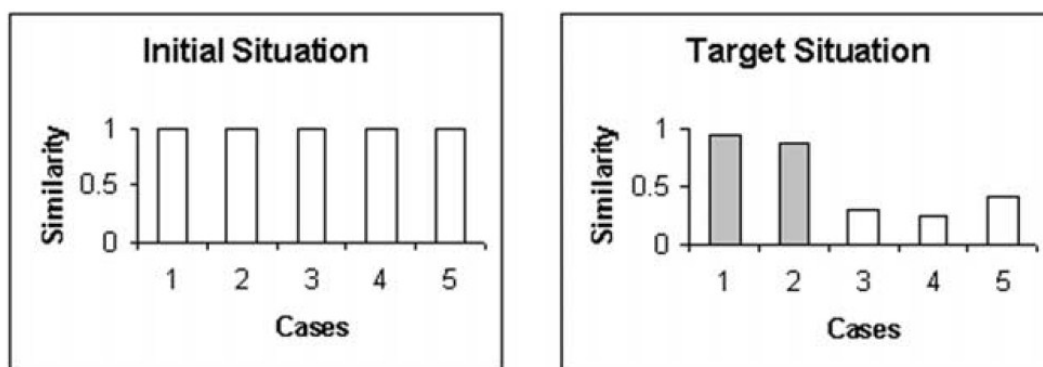
Atribut **k uvolnění (relaxation)** je vybrán jako atribut s nejvyšší entropií (nejnižším informačním ziskem) mezi atributy, které uživatel použil k omezení.

Problémem této techniky je náročnost výpočtu entropie, protože vypočítáváme zisk pro všechny parametry. Pro více informací odkážu laskavého čtenáře na [22].

**Metoda vyplývání** do jisté míry řeší problémy entropických technik. Nespoléhá se na ideálního uživatele a primárním cílem už není maximálně snížit počet otázek, což by vedlo ke snížení kvality dialogu.

Místo toho, abychom zjišťovali informační zisk, zvážíme podobnostní rozdělení kandidátských případů. To má zásadní výhodu zejména v měření podobnosti. Můžeme totiž při výpočtu zahrnout váhy, a to i uživatelské.

Při inicializaci jsou všechny případy shodně podobné se zadáním, a proto nerozlišitelné. S každým dalším parametrem dotazu se podobnost mění. Takto v iteraci zvyšujeme podobnost až k cílové sadě.



Obrázek 10 - Proces podobnosti případů [simVar]

Problémem je náročnost předpovědi podobnosti pro další kroky výběru. Popsanou techniku využili autoři například v projektu simVar [23].

### 4.3.2 Metoda průběžných doporučení

Většina metod založených na průběžném doporučování je vhodná pro systémy, kde má uživatel nízkou znalost problémové domény. Například metoda preferencí už nepředpokládá téměř žádnou znalost.

Možnosti zpětné vazby:

- **Zadání hodnot hledání (value elicitation)** – očekává se explicitní zadání vyhledávacích parametrů
- **Hodnocení (rating based)** – opět explicitní zadání uživatele, který ohodnocuje jednotlivé položky.
- **Kritikou (critique based)** – [24] Na závěr doporučení uživatel rozhodne, zda pro něj mělo nějaký přínos, případně míru přínosu. Uživatel může kritizovat přímo vybrané parametry. Například: Další produkt by měl být levnější. Další produkt by měl být v dřívějším termínu. Samostatné použití této metody může vést k velmi dlouhým konverzacím, než systém nasbírá dostatek informací obdobně jako další metoda.
- **Preferencí (preference based)** – uživateli je nabídnuto několik výrobků a ten rozhodne, který z nich pro něj má největší význam. Tato metoda je velmi vhodná pro uživatele s velmi nízkou znalostí domény. Ti se sice nevyznají v parametrech, ale jsou schopni intuitivně rozhodnout o preferenci. Problém je zjistit, proč se který uživatel rozhodl pro který produkt. Dokonce tak mohlo dojít na základě atraktivní fotografie.

S technikami průběžného doporučení je velice často spojován také pojem *Různorodosti výsledků*. Jeho využití může významně ovlivnit počet iterací komunikace.

#### Různorodost výsledků

Těžko budeme zjišťovat zpětnou vazbu, pokud uživateli nabídneme několik stejných produktů jen proto, že byly nejpodobnější jeho požadavkům. Pro uživatele samotného nemají taková doporučení vysokou hodnotu. Proto je dobré zamyslet se nad různorodostí výsledků a do top pozic vybírat navzájem méně podobné produkty. Uživatel tak má větší manévrovací možnosti se rozhodnout.

Existuje několik technik jak toho dosáhnout. Produkty se mohou načíst náhodně, to ale není příliš efektivní. Také je možné produkty rozdělit do skupin podle hladiny podobnosti a vypsát poté jen jednoho kandidáta z každé skupiny.

Případně je možné použití nějaké sofistikovanější metody, jako je například bouded greedy technique – využívá iteračního přístupu načítání výsledků a seskládání výsledného seznamu k doporučení. V [25] se uvádí zvýšení rozmanitosti o 50 % při jen 10 % snížení podobnosti od požadavků.

Hladinový výpis výsledků je také vhodný pro vysvětlení doporučení. V každé skupině můžeme zvýraznit nebo jinak okomentovat důvod, proč je tato skupina produktů doporučena.

Bližší informace a popis několika technologií naleznete v [26]

### 4.3.3 Hybridní řešení

Hybridní řešení kombinuje více komunikačních technik. Například ExpertClerk využívá v začátku metodu otázek a později jemněji upřesňuje doporučení pomocí kritiky. Použití více technik je obecně doporučováno. Dosahuje se vyšší úspěšnosti doporučení a hlavně zkrácení délky konverzace. Například pokud využijeme jen metodu průběžných doporučení, bude zapotřebí mnoho iterací, než zjistíme opravdové preference zákazníka. V kombinaci s komunikací metodou otázek ale můžeme zjistit množství informací dopředu a metodou průběžných doporučení jen zjemňovat preference. Obecně je uživatel více ochoten odpovídat na otázky na začátku relace než v jejím průběhu.

## 4.4 Personalizace

V kapitole 4.1 jsem zmínil termín *personalizovaný*. Nyní vysvětlím jeho význam. Jedna z definic je dostupná zde [27].

*Personalizace umožňuje vkládání, přizpůsobení nebo návrh obsahu v jakékoliv podobě, která je relevantní pro každého uživatele. Zakládá se na implicitním chování, preferencích a explicitně získaných datech.*

Personalizace tedy vede k přizpůsobení danému uživateli. Abychom toho dosáhli, musíme mít možnost uživatele identifikovat a sbírat o něm informace – vytvářet uživatelský profil. Konkrétně u webových aplikací je většinou identifikace realizována pomocí přihlašování, případně v rámci relace uložené pomocí cookies. V tomto ohledu dělíme personalizaci na:

**Silná personalizace** - Pokud je vytvořen uživatelský profil, který nese dlouhodobé preference.

**Slabá personalizace** - V rámci jedné relace je vytvořen profil uživatele, ten ale nemá dlouhého trvání, zpravidla jen po dobu života relace. Informace i tak mohou být archivovány, jejich využití je pak pro podobného uživatele do budoucna.

### 4.4.1 Kontext

Velmi důležitým faktorem při doporučování je kontext. Jde o souvislost s aktuálními potřebami. V doméně prodeje zájezdů má kontext velký význam, protože nákup dovolené je zpravidla sezónní záležitost s velkým odstupem opakování. U jednoho uživatele se mohou preference lišit například ve chvíli, kdy objednává eurovíkend nebo dovolenou s rodinou. Kontext může výrazně ovlivnit váhy preferencí.

### 4.4.2 Profilování

Profilování je získávání informací za účelem vytvoření uživatelského profilu. V základu může být tvořen z demografických údajů – například věk, bydliště, pohlaví. Ty pak mohou být využity jako rozšiřující informace doporučení. Profil sestavujeme pomocí explicitní a implicitní zpětné vazby.

#### Explicitní zpětná vazba

Explicitní zpětná vazba je takovou formou informace, kterou vědomě zadává uživatel. Nejčastěji se tak děje pomocí hodnocení položek nebo samotného doporučení - například počet hvězdiček nebo známky jako ve škole. Problém je v tom, že je to pro uživatele krok navíc, ze kterého

není na první pohled viditelný přínos. Výzvy k hodnocení jsou proto opomíjeny a systém se tak velice pomalu nebo špatně učí. Lidé chtějí pokud možno maximální usnadnění - automatizaci.

Takovou nejznámější nepovedenou ukázkou pokusu o získávání explicitní zpětné vazby je SearchWiki [28] od společnosti Google. Ten umožňoval řazení výsledků podle uživatelských preferencí. Ty, které neodpovídaly představám a zadání, bylo možné přesunout na konec nebo úplně odstranit, ty odpovídající naopak přesunout na vrchol stránky. Doplnkem byla také služba SideWiki, kde uživatel mohl výsledky hledání komentovat. Bohužel služby nenaplnily očekávání a Google je byl nucen stáhnout.

V projektu využijeme explicitní zpětnou vazbu ve formě získávání informací o uživatelských preferencích pomocí jednoduchých konverzačních technik.

### **Implicitní zpětná vazba**

Z chování zákazníka můžeme v průběhu času poskládat jeho profil, bez toho abychom jej zatěžovali sadou otázek případně jinou formou výzvy ke zpětné vazbě. V rámci iterací relace je nejdůležitější částí výběr preferované nabídky z prezentovaných dat, a to při několika různých příležitostech s různou mírou důležitosti. Jsou to například:

- objednání zboží,
- opakované navštívení,
- uložení produktu do seznamu přání,
- tisk nebo odeslání kamarádovi,
- míra opuštění stránky a čas věnovaný jedné nabídce.

Nejvyšší vypovídací hodnotu má úspěšné zakončení výběru. U eshopu to je objednání zboží, nejlépe s uskutečněným prodejem.

### **Faktory negativně ovlivňující profilování**

**Změna preferencí uživatele** – uživatel se v čase vyvíjí a stejně tak se vyvíjí jeho preference. Tím více u produktů občasného nákupu. Například pokud uživatel založí rodinu, dá se předpokládat, že bude nakupovat jinak zaměřenou dovolenou.

**Náhodná listování** – registrovaný uživatel si může listovat v katalogu jen pro „rozšíření obzorů“, nikoliv ze skutečného zájmu. Například pokud se bude chtít podívat kolik stojí dovolená na kterou jel jeho souseď.

**Výměna uživatele** – například v situaci, kdy registrovaný uživatel nakupuje pro někoho jiného. Pak se dá těžko předpokládat, že průběh takového hledání bude dobrým zdrojem preferencí registrovaného uživatele. Tuto skutečnost nejsme schopni ovlivnit.

### **Přídavné parametry**

Až doposud jsme předpokládali, že dotazy se skládají ze stejných atributů, které jsou použity v popisech produktu. Ale v CBR učícím se systému můžeme doplnit jakékoliv vlastnosti pro zjemnění, které ani nejsou v sadě vlastností produktu evidovány. Slouží pak jen k rozlišení případů, ne však pro

vyhledání položek přímo v databázi. Takovým přídatným parametrem může být například „Vyžadují klidnější destinaci“, databáze ovšem takovou informaci nedisponuje.

## 4.5 Shrnutí

V aktuální kapitole jsem nastínil problém doporučování informací. V krátkosti jsem je rozdělil do několika proudů a podrobněji popsal metodu CBR, která bude nejlépe vyhovovat potřebám projektu, a to zejména z pohledu orientace na kontext a eliminaci problému studeného startu i s ohledem na uživatele, který v zamýšlené doméně hledá informace sezónně.

V části 4.3 jsem vysvětlil výhody hledání informací pomocí dialogu oproti single-shot doporučení a také techniky pro jeho dosažení. Bohužel všechny uvedené metody mají pro můj projekt nepřekonatelné vady. Postupy výběru otázek, uvedené v této kapitole, jsou příliš výpočetně náročné a pro předpokládaný počet položek báze jsou nepoužitelné. Například metoda průběžných doporučení by vedla k velmi dlouhým relacím, které by uživatel jen těžko dovedl ke zdárnému konci.

Zejména pro mobilní zařízení je přesto použití komunikační metody nutností, a to z několika důvodů. Primárně potřebujeme vhodně omezit sadu kandidátských položek k doporučení, a to takovým způsobem, abychom uživatele zbytečně nenutili zadávat množství hodnot do rozsáhlého formuláře, a to hlavně s ohledem na uživatelské rozhraní.

Je také vhodné uvažovat nad různorodostí výsledků ve spojení s vysvětlením doporučení. Uživatel tak má větší manévrovací prostor k rozhodování a také je srozuměn s automatickými úpravami dotazu a důvody doporučení. Vhodné zobrazení všech těchto informací na limitovaném prostoru mobilního zařízení je námětem k dalšímu zkoumání.

V poslední části jsem se zmínil o personalizaci, profilování a kontextu. Nákup dovolené je sezónní záležitostí s velmi řídkou frekvencí opakování. Trvalo by velice dlouho sesbírat dlouhodobou preferenci uživatele, navíc s vysokým rizikem změny této preference v průběhu času. Potřeby se také liší podle druhu dovolené. Z tohoto důvodu převažuje kontext nad silnou personalizací. Identifikace uživatele bude dostupná a úspěšné relace budou ukládány pro pozdější využití CBR, ale získané preference budou mít význam převážně jen v rámci aktuální relace. Takto budeme schopni doporučovat s nejvyšším ohledem na kontext s využitím zkušeností jiných uživatelů. V tomto ohledu můžeme mluvit o kolaborativním přístupu.



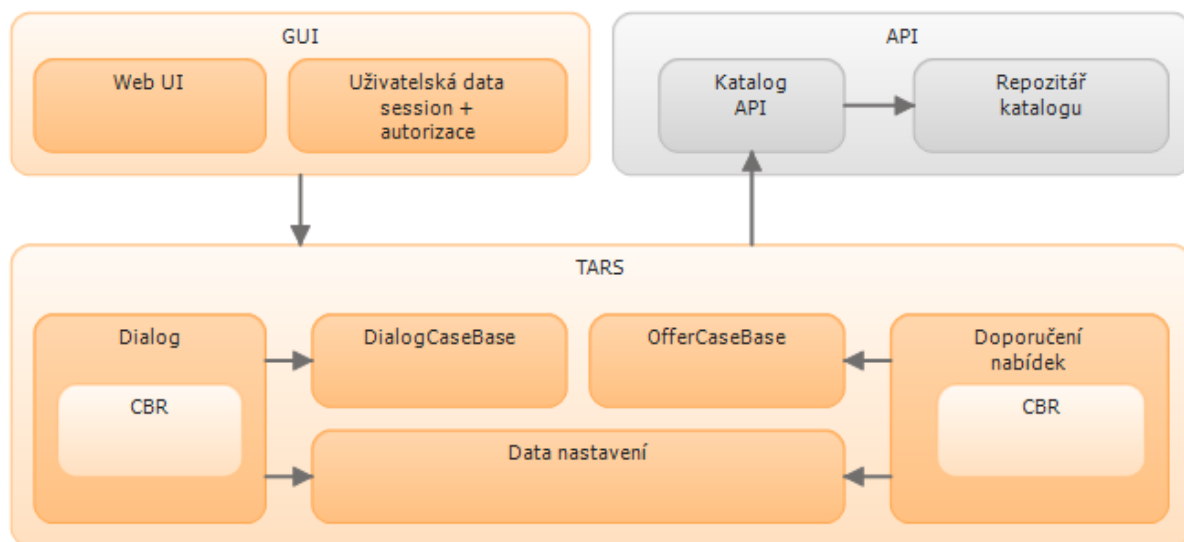
## 5 Aplikace doporučování zájezdů - TARS

V předchozích kapitolách jsem nastínil problematiku vývoje pro mobilní zařízení, popsal jsem možnosti doporučování informací/produktů, blíže jsem se zaměřil na technologie CBR (viz kapitola 4.2). Také jsem uvedl doménu cestovního ruchu, stávající stav a potřebu jeho řešení. Všechny uvedené informace v této kapitole prakticky aplikuji do prototypové aplikace doporučování zájezdů pro mobilní zařízení. Pracovní název projektu je TARS – Travel Agency Recommender System. Cílem kapitoly je popis aplikace TARS zejména z pohledu doporučování informací, posuzování podobností a učících algoritmů.

### 5.1 Architektura

Ve firmě, kde pracuji, vzniká software určený převážně pro cestovní ruch. Mnoho komponent je již vyvinuto. TARS je součástí mnohem většího systému, proto nebylo potřeba vyvíjet komponentu zajišťující data o zájezdech. Využívám původní a již hotové součásti firemního systému, a to API katalogu. Pro lepší orientaci přikládám obrázek 11 obecného pohledu na TARS a jeho části. Patrně jsou tři základní komponenty:

- **TARS** – logika celého systému doporučování
- **GUI** – grafické rozhraní potřebné pro komunikaci s uživateli
- **API katalogu** – webová služba zajišťující data o zájezdech



Obrázek 11 – Přehledové schéma - části systému TARS.

**API katalogu** se stará mimo jiné o komunikaci s bázi dat o zájezdech a jejich získávání, včetně dataminingu vybraných informací. Popis vnitřního chování není předmětem této práce, a proto se k němu vrátím jen výjimečně z důvodů osvětlení funkce vyvíjených komponent. API je přístupné jako webová služba a je jediným zdrojem detailních i souhrnných informací o zájezdech včetně ověřování jejich počtu. API neobsahuje jakoukoliv podporu pro doporučování informací a ani sběr dat

k tomu potřebných. Proto TARS vznikl jako samostatná vrstva obsahující také repositáře pro správnou funkci CBR.

**TARS** je hlavním částí systému, pro svou funkci využívá dvě fáze doporučení:

- část dialogová – zabezpečuje získání a sestavení vhodného dotazu,
- část doporučení – na základě dotazu doporučí a nabídne zájezdy.

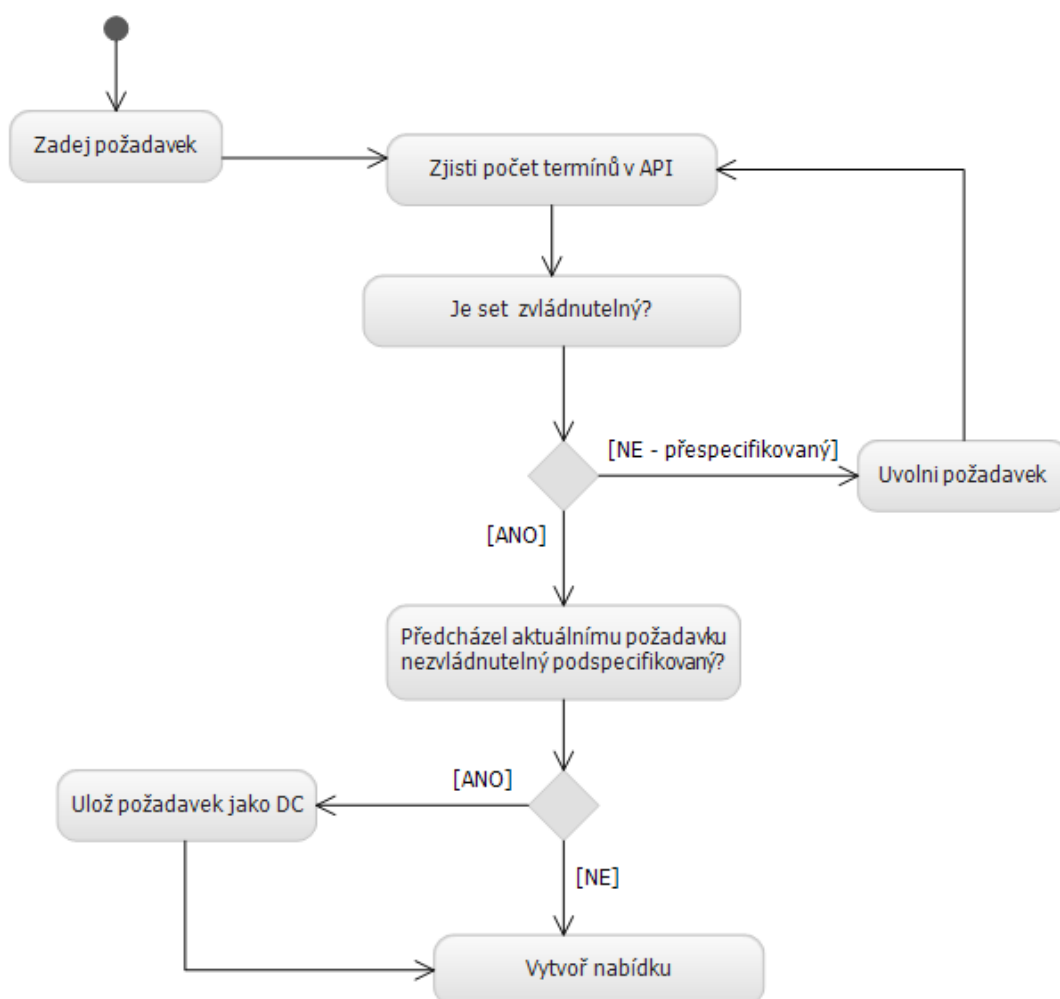
Blíže je rozeberu v následujících kapitolách.

**GUI** je rozebráno v samostatné kapitole (viz kapitola 5.5)

## 5.2 Část dialogová

Je zodpovědná za získání informací o preferencích uživatele. Cílem je sestavit vhodný dotaz pro další část doporučení. V závislosti na aktuálním zadání koriguje sestavování dotazu, nebo jej sám upraví, a to buď uvolňováním nebo další komunikací s uživatelem vhodnou sadou otázek.

Systém je schopen využívat předchozích zkušeností uživatelů, kteří již komunikovali v minulosti, ale také těží informace daného uživatele. Názorně v diagramu aktivit viz obrázek 12.



Obrázek 12 – Diagram aktivit – část dialogová

Před rozkladem diagramu musím ještě vysvětlit pojem zvládnutelnost setu. Je nutné nezapomenout na to, že pracujeme nad daty v rozsahu milionů záznamů. Samotný proces doporučení zahrnuje několik operací výpočtu podobnosti položek. Pokud by měl systém aplikovat tyto algoritmy pro všechny položky báze, uživatel by se odpovědi nedočkal. Nehledě na to, že systém musí být dimenzován na množství souběžně přistupujících uživatelů. Proto před samotným doporučovacím procesem omezíme sadu nabídek, nad kterými budeme vypočítávat podobnosti. Opačným problémem je velmi malé až nulové množství nabídek, nad kterým již nemá doporučování smysl. Pro oba extrémy jsou nastaveny prahové hodnoty. Cílem dialogové části je sestavit takový dotaz, který po aplikaci získá počet nabídek mezi nastavenými hranicemi. Pak je takový set aplikací zvládnutelný pro algoritmy doporučování.

Nyní již k diagramu aktivit viz obrázek 12. Uživatel zadá hodnoty vlastností dle svých preferencí. Ty jsou odeslány do API, které zjistí pro aktuální dotaz počet dohledatelných nabídek v databázi. Následují tři možné scénáře.

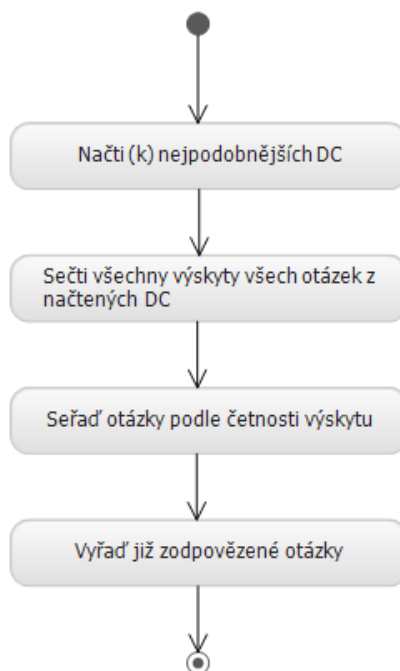
- **Zvládnutelný set** - počet nabídek v setu je v nastavených mezích.
- **Nezvládnutelný pod-specifikovaný set** - počet nabídek převyšuje horní nastavenou hranici.
- **Nezvládnutelný pře-specifikovaný set** - počet nabídek je pod dolní nastavenou hranicí nebo je nulový.

### 5.2.1 Zvládnutelný set

Pokud je set zvládnutelný, pokračuje rovnou dál k části doporučení.

### 5.2.2 Nezvládnutelný pod-specifikovaný set

Pokud API zjistí, že výsledný set přesahuje nastavenou hranici počtu termínů, je set nezvládnutelný pod-specifikovaný a my potřebujeme dotaz dále omezit. V takovém případě dialogový systém musí uživatele vyzvat k dalšímu zadání. Abychom jej obtěžovali co nejméně a zároveň mu v maximální míře pomohli s vyplňováním hodnot vlastností, využijeme systém CBR respektive jeho části SBR – similarity based recommendation (bližší informace [29]). Doporučíme vhodnou sadu otázek podle zkušeností předchozích uživatelů.



Obrázek 13 - Výběr seznamu otázek

Celý proces je znázorněn na obrázku 13. V první fázi vypočítám podobnosti všech DialogCase – DC z DialogBase (báze již naučených případů dialogu) s aktuálním požadavkem. K tomu využijeme výpočet míry podobnosti Dialog Case Similarity Measure. Výpočet se řídí pravidly a mírami nastavenými doménovým expertem. Z výsledků vybereme  $k$  nejpodobnějších případů.

V druhé fázi postupně projdu všechny případy z předchozí fáze a sečtu všechny výskyty otázek obsažených v řešeních. Jako výsledné skóre nám bude dostačovat i tento počet výskytů. Zjemnit výpočet je možné využitím míry podobnosti z předchozí fáze a výskyty otázek opatřit vahou.

Podle počtu výskytů otázky seřadíme a vyloučíme ty, na které již uživatel odpovídal při předchozí iteraci. Poté otázky vypíšeme v grafice.

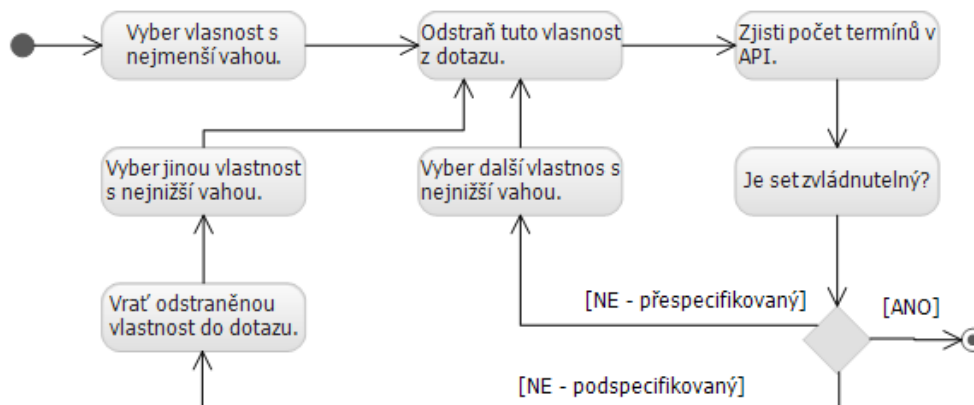
První fáze využívá data z DialogCaseBase, ta jsou sbírána na základě úspěšné specifikace dotazů, které vedly k zvládnutelnému setu. V diagramu na obrázku 12 jde o část algoritmu od kladného rozhodnutí o zvládnutelném setu. Pokud uživatelův požadavek nebyl zvládnutelný, systém si zapamatuje tento stav a pokračuje v dialogu s klientem. Pokud se úpravou požadavku stane set zvládnutelným, systém zjistí, které otázky vedly k tomuto úspěchu. Nakonec sestaví DC z původního nezvládnutelného požadavku (v roli problému) a seznamu otázek (jako řešení). Tento DC uloží do DialogCaseBase pro další využití.

### 5.2.3 Nezvládnutelný přespecifikovaný set

Pokud API vrátí počet nabídek, který je pod spodní hranici zvládnutelnosti, nebo je dokonce nulový, mluvíme o nezvládnutelném přespecifikovaném setu. Mohli bychom uživatele vyzvat k další interakci a nechat jej uvolnit některé požadavky, nebo, jako v single-shot, zahájit úplně nové hledání.

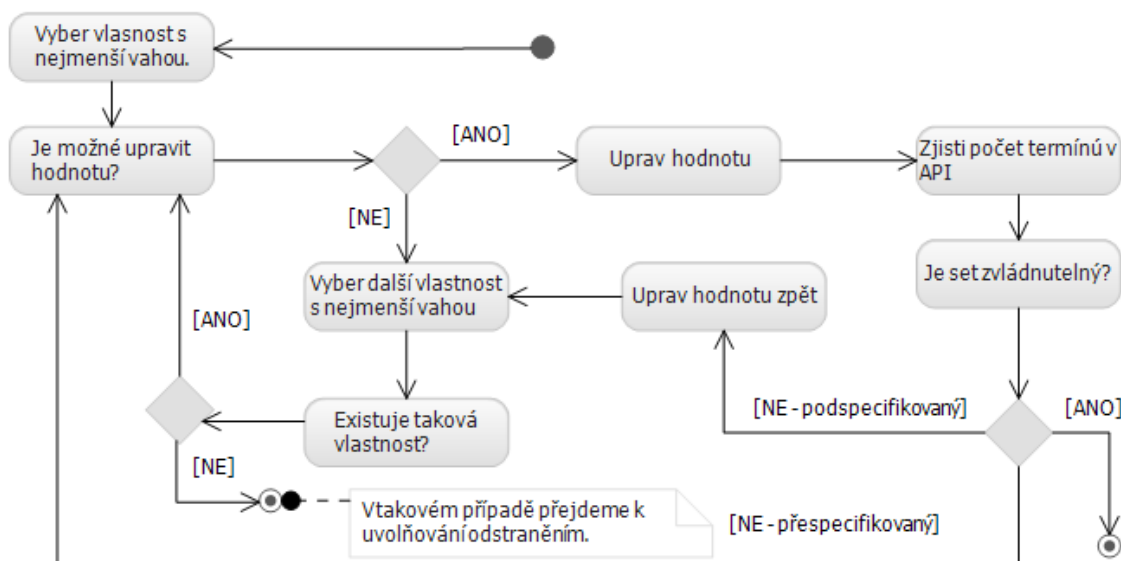
Nebo naopak vynecháme krok další interakce a rovnou automaticky uvolníme některé vlastnosti požadavku a zkusíme vyhledat podle nově sestaveného dotazu. Tento postup se nazývá query-relaxation (obecný popis viz kapitola 4.3).

V prototypovém provedení je využito uvolňování vlastnosti jejím úplným vyloučením. Vybrána je vlastnost s nejmenší vahou. Algoritmus je znázorněn na obrázku 14. Po každé změně ověřujeme, zda úprava měla vliv na zvládnutelnost setu. Pokud ne, postup opakujeme. Každé uvolnění vlastnosti je zaznamenáno a v rozhraní oznámeno uživateli.



Obrázek 14 – Uvolňování přespecifikovaného požadavku odstraněním atributů s nejmenší vahou

V další verzi bych rád využil sofistikovanější algoritmus uvolňování pomocí úpravy hodnot. Jeho diagram je na obrázku 15. Postup je komplikovanější v tom, že jsou nastavena pravidla pro změny hodnot. Jsou určeny hranice, kam až můžeme hodnotu posouvat a také krok, v jakém bude úprava aplikována. Opět po každé změně ověříme v API, zda je nový dotaz zvládnutelný. Pokud ne, pokračujeme v úpravách. Pokud ovšem vyčerpáme všechny vlastnosti a set je stále nezvládnutelný, je potřeba použít jiný postup. Například přistoupit k algoritmu uvolňování odstraněním, nebo rovnou vyzvat uživatele k manuální úpravě požadavku.



Obrázek 15 - Uvolňování požadavku úpravou hodnoty

Oba algoritmy jsou řízeny pravidly nastavenými doménovým expertem a pomocí vah. Pro uživatele bez profilu a s prázdnou relací jsou váhy přednastaveny. Váhy průběžně získáváme profilováním uživatele jak dlouhodobým tak krátkodobým – v rámci jedné relace (viz Profilování - 5.2.5).

#### 5.2.4 Dialog case – DC

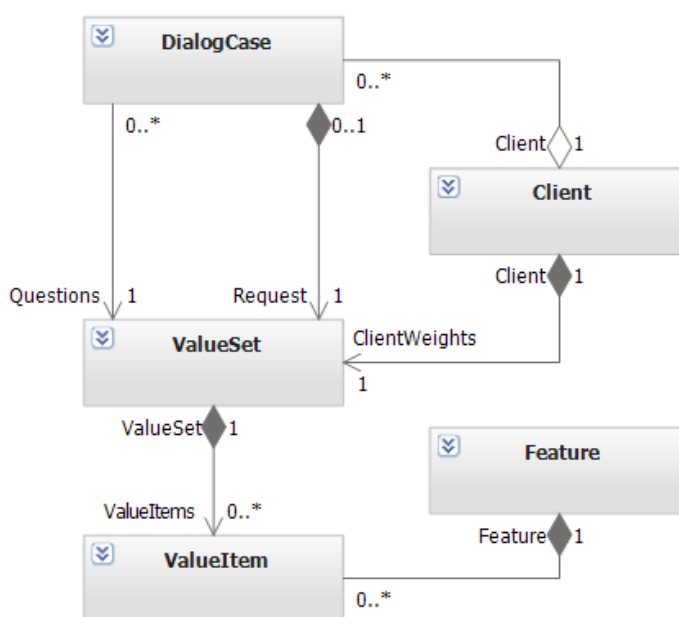
Dialog case je objekt sloužící k uložení případu pro CBR dialogové části. Dle třídního diagramu na obrázku 16, se DC skládá primárně z požadavku a sady otázek. V terminologii CBR:

**Problém** – je požadavek (Request), který původně vedl k nezvládnutelnému setu a nutnosti dialogu ve formě doporučení dalších otázek.

**Řešení** – je seznam otázek (Questions), které dotaz dovedly k zvládnutelnému setu.

Požadavek i seznam otázek je v systému reprezentován objektem ValueSet, který je složen z ValueItems. Ty uchovávají právě hodnoty vah i hodnoty atributů a mají důležitou vazbu na objekt Feature, který nese informaci o typu atributu a jeho nastavení.

Vazba klienta s DC umožní další využití v případě získávání dlouhodobých preferencí a vzorců chování uživatele. Důležitou vlastností klienta je ClientWeights - uložisko uživatelských vah. Ty jsou hlavním nositelem uživatelského profilu.



Obrázek 16 – Zjednodušený třídní diagram – DialogCase

#### 5.2.5 Profilování

Protože v části doporučení nemá uživatelský profil význam, budu se jím zabývat výhradně pro část dialogu. Profilování má v TARS několik významů. V rámci relace vytváříme dotaz - požadavek, který je použit pro vyhledávání a doporučení. Doplněkem požadavku jsou váhy vlastností. Ty jsou využity jak v rámci relace, ale hrají roli i pro dlouhodobý profil uživatele.

**Dotaz - požadavek** – V průběhu relace uživatel vyplňuje hodnoty vlastností požadavku. Ty jsou zaznamenávány a slouží k sestavení dotazu pro další část systému. Tyto hodnoty mají význam jen

po dobu trvání relace, po jejím vypršení nejsou dále uchovávány. Požadavek může být upravován i automaticky, jak jsem již zmínil v kapitole 5.2.

**Váhy** – Váhy jsou implicitně nastaveny podle doménového experta. Upravovány jsou podle chování uživatele. TARS pro výpočet uživatelských vah používá dva zdroje.

- Učení vah podle požadavků.
- Učení vah podle rozdílnosti požadavku a nabídky.

**Učení vah podle požadavků** – vycházím z jednoduchého předpokladu, že pro uživatele, který při každém zadání požadavku vyplní například hodnotu atributu „Počet dní“, je tento atribut důležitý. Naopak atribut, který nevyplnil nikdy, pro něj takový význam nemá. Proto váhu atributu, který uživatel poslal jako součást požadavku do systému, inkrementuji. Váhy atributů, které nevyplnil, dekrementuji.

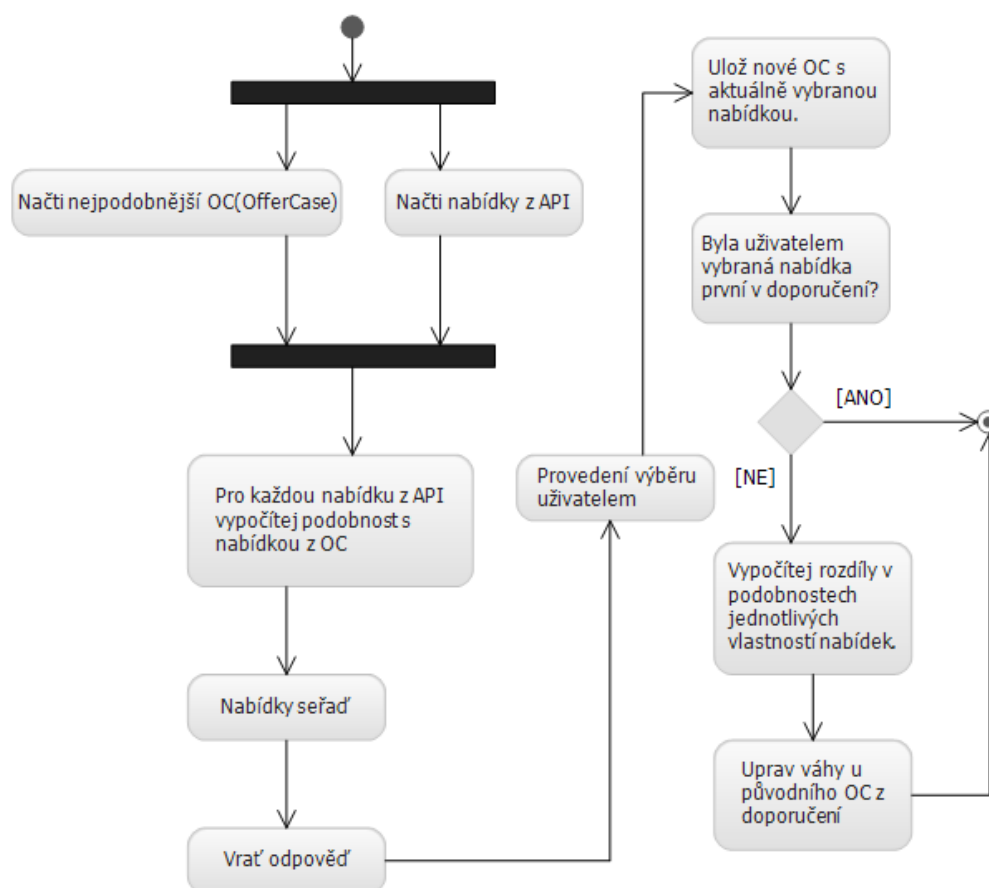
**Učení vah podle rozdílnosti požadavku a nabídky** – TARS byl v prototypu nastaven tak, že atributy požadavku a atributy nabídek jsou navzájem až na drobné odlišnosti srovnatelné. Proto jsme schopni vypočítat alespoň orientační podobnost požadavku a nabídky. V takovém případě můžeme použít také algoritmus učení z kapitoly Proces učení vah (viz 5.3.1) s tím rozdílem, že podobnosti prvního a dalších nabídek jsou vypočítány ve vztahu s požadavkem.

## 5.3 Část doporučení

Část doporučení je druhou z hlavních částí TARS. Je jedním z hlavních cílů práce a slouží k selekci nabídek podle potřeb uživatele, které jsou pak zobrazeny k výběru. Tato část již naplno využívá systém CBR a učí se z výběru předchozích hledání. A to nejen daného uživatele, ale napříč všemi zkušenostmi ostatních uživatelů. Hlavním důvodem je kontext popsáný v kapitole 4.4.1. Cílem je dosáhnout takového doporučení, které přesně odpovídá přáním uživatele a taková doporučení si uložit a použít v dalších hledáních.

Vstupem je již složený dotaz z dialogové části, na jehož základě bude probíhat doporučení. Celý proces je popsán na aktivním diagramu na obrázku 17.





Obrázek 17 – Diagram aktivit – Průběh doporučování nabídek

Po získání dotazu načteme potřebná data pro výpočty podobnosti. Na jedné straně to jsou kandidátské nabídky - termíny zájezdů. Ty získáme z API katalogu. Na druhé straně načteme nejpodobnější OfferCase, který nejvíce odpovídá aktuálnímu dotazu. Podobnost se vypočítává pomocí přednastavených pravidel a vah (viz kapitola 5.4).

Dále systém porovná všechny získané kandidátské nabídky s nabídkou uloženou v nejpodobnější OC. Tím získáme skóre každé nabídky a podle něj je můžeme seřadit. Takto připravená data vypíšeme uživateli.

Pokud mu vyhovuje některá z nich, vybere ji a tím nám dává najevo, že doporučení bylo prospěšné. Toto jsou důležité informace, které uložíme jako OC pro využití v budoucnu. Abychom doporučení ještě vylepšili, zjistíme, zda uživatelem vybraná nabídka byla první ze sady doporučených. Pokud ne, můžeme se naučit, které vlastnosti jsou v daném kontextu pro uživatele důležité. Tento algoritmus podrobněji popíšeme v další kapitole.

### 5.3.1 Proces učení vah

Abychom zajistili jemnější výpočet podobnosti nabídek, využijeme postup učení vah. V modelovém případě jsme uživateli zobrazili několik nabídek v setu. Ten si vybral druhou v pořadí. Z toho můžeme usuzovat, že první doporučení nebylo úplně dle jeho představ. Z rozdílnosti podobnosti jednotlivých vlastností nabídek setu jsme schopni určit, která z vlastností má pro uživatele větší nebo menší význam v konkrétním kontextu. Pro názornost použijeme příklad dle tabulky 2.

Vlastnost	První nabídka v setu		Druhá nabídka v setu		Rozdíl podobnosti	Vliv na váhu vlastnosti
	Hodnota	Podobnost	Hodnota	Podobnost		
Destinace	Řecko	0,8	Itálie	0,2	0,6	Dekrementace
Cena	5000	0,5	2000	1,0	-0,5	Inkrementace
Bazén	ano	1,0	ano	1,0	0,0	Stagnace

*Tabulka 2 - Výpočet pro určení atributů vhodných pro učení vah*

V tabulce jsou zobrazeny dvě nabídky setů. Druhou nabídku uživatel vybral, a proto je možnost se naučit váhy. Ve sloupcích „Podobnost“ je hodnota podobnosti jednotlivých vlastností vzhledem k nabídce uložené v OC. Hodnoty podobnosti z druhé nabídky odečteme od hodnot první (systémem preferované). Nyní máme ve sloupci „Rozdíl podobnosti“ vypočteny rozdíly, podle kterých určíme nové hodnoty vah vlastností. Pokud je hodnota menší než nula, znamená to, že tato vlastnost měla pro výběr pravděpodobně vyšší váhu. Proto také inkrementujeme váhu uloženou v OC. Obdobně postupujeme při kladné hodnotě, jen opačně váhu dekrementujeme. Hodnotu koeficientu inkrementace a dekrementace bylo vhodné nastavit na 1,2 (viz kapitola 6.1.2). Inkrementace je dána vztahem:

$$w = w * k ,$$

kde  $w$  je váha a  $k$  je koeficient.

Váhu uložíme k OC až po procesu doporučení, takže vliv bude mít až na hledání v budoucnosti a aktuální doporučení již neovlivní.

### 5.3.2 Offer case – OC

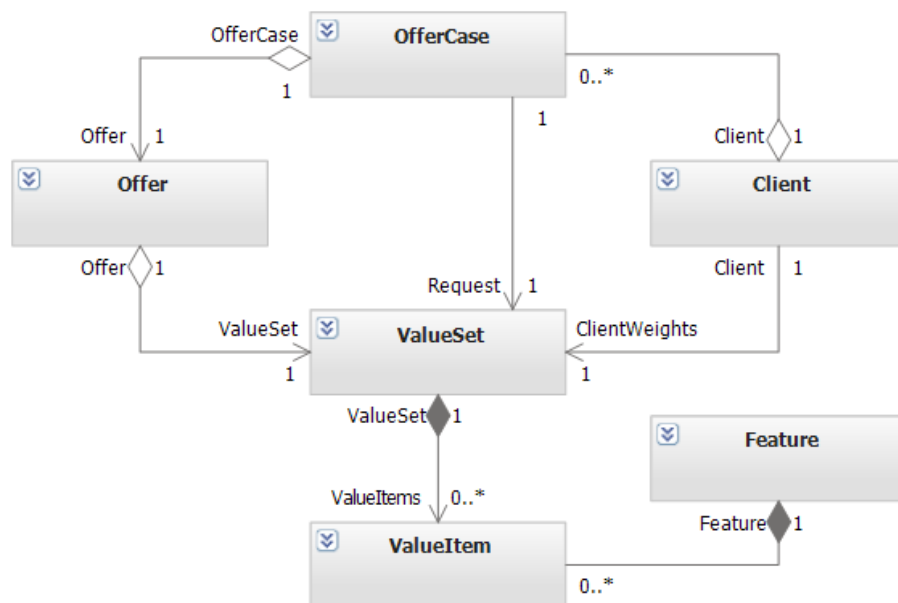
Offer case je objekt sloužící k uložení případu CBR části doporučení. Pro základní CBR jsou klíčové objekty Offer a Request.

**Problém** – je požadavek (**Request**), který popisuje preference uživatele.

**Řešení** – nabídnutý a akceptovaný zájezd (**Offer**).

Požadavek i nabídka je v systému reprezentována objektem ValueSet, který je složen z ValueItems. Ty uchovávají právě hodnoty vah i hodnoty atributů a mají důležitou vazbu na objekt Feature, který nese informaci o typu atributu a jeho nastavení.

**Client** slouží k identifikaci uživatele, pro kterého byl offercase vytvořen. Později budeme schopni těžit informace o hledání klienta.



Obrázek 18 - Class Diagram - OfferCase

## 5.4 Posuzování podobnosti

V TARS využíváme CBR a SBR, jejichž základem jsou výpočty podobnosti ( $\text{sim} = \text{similarity}$ ), a to jak pro fázi hledání nejpodobnějších případů tak pro fázi porovnávání kandidátských případů. Dá se říci, že posuzování podobnosti je nejdůležitější část TARS. Výpočet vychází ze vzorce:

$$\text{sim}(t, c) = \frac{\sum_{i=1..n} w_i \times \text{sim}(t_i, c_i)}{\sum_{i=1..n} w_i}$$

$\text{sim}(t, c)$  je podobnost položky  $t$  a  $c$ ,  $n$  je počet vlastností,  $w$  je váha vlastnosti,  $\text{sim}(t_i, c_i)$  je podobnost vypočítaná pro vlastnosti  $i$  mezi položkami  $t$  a  $c$ .

Ze vzorce jsou patrné dvě důležité položky:

- **Váha vlastnosti** – říká, jak moc významná je vlastnost pro hodnocení. Nastavena je doménovým expertem, může být upravována profilem nebo učením vah viz kapitola 5.3.1
- **Výpočet podobnosti** pro danou vlastnost – je závislý na typu podobnosti.

Nyní blíže rozeberu jednotlivé typy vlastností.

### 5.4.1 Binární vlastnosti

U vlastností nabývajících jen hodnot `true` nebo `false` jednoduše nastavíme hodnotu podobnosti 1, pokud odpovídají, a 0, pokud neodpovídají.

### 5.4.2 Pole binárních vlastností

Pro velké množství vlastností s binárním typem je vhodné použít metodu jaccard dle Tanimota. Sčítáme shody a výskyty podle vzorce:

$$\text{sim}(x, y) = \frac{\sum_i (x \cap y)}{\sum_i (x \cup y)}$$

Kde  $\text{sim}(x, y)$  je podobnost pole binárních vlastností  $x$  s polem  $y$ . Zlomek se dá interpretovat jako podíl součtu souhlasně ohodnocených bitů vlastností a součtu rozdílně ohodnocených bitů.

### 5.4.3 Číselné vlastnosti

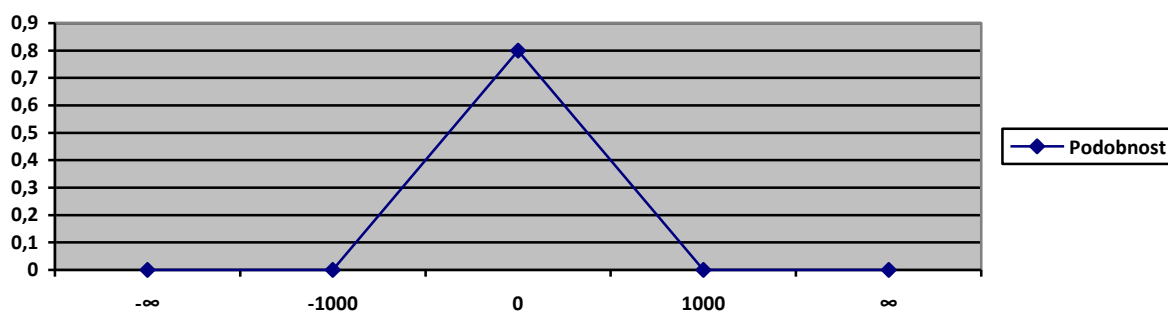
U číselných typů je rozhodující rozdíl hodnot, pro který rozlišují *hladinové* nebo *lineární* posouzení podobnosti.

U **hladinového** můžeme nastavit intervaly, pro které bude přesně nastavena hodnota podobnosti. V tabulce 3 je konkrétní příklad hladinové nesymetrické podobnosti. Pro hodnoty lišící se od 0 do 1000, bude nastavena podobnost 0,4. Pro hodnoty lišící se ve stejném intervalu (ale záporně), bude podobnost 0,8. Mimo tyto intervaly posuzujeme rozdíly hodnot jako nepodobné. Například pro dvě podle tabulky posuzované hodnoty 5000 a 4800, je nastavena podobnost 0,4.

Dolní hranice	Horní hranice	Podobnost
$-\infty$	-1000	0
-1000	0	0,8
0	1000	0,4
1000	$\infty$	0

Tabulka 3 - Hladinové určení nesymetrické podobnosti

**Lineární** podobnost má podobné zadání. Jen hodnota podobnosti není konstantní pro celý interval, ale platí pro hraniční hodnotu. V grafu (viz obrázek 19) je konkrétní příklad lineární symetrické podobnosti.



Obrázek 19 - Graf lineární symetrické podobnosti

$$d(x, y) = \left(1 - \frac{x - y}{r}\right) * d_r$$

Kde  $r$  je rozsah vlastnosti pro rozhodný interval,  $d_r$  je nastavená podobnost pro rozhodný interval a hraniční hodnotu. Například výpočet podobnosti podle grafu a hodnoty 5000 a 4800 vypočítáme takto:

$$d(5000, 4800) = \left(1 - \frac{5000 - 4800}{1000}\right) * 0,8$$

$$d(5000, 4800) = 0,8 * 0,8 = 0,64$$

Výsledná podobnost pak je 0,64.

#### 5.4.4 Datumové vlastnosti

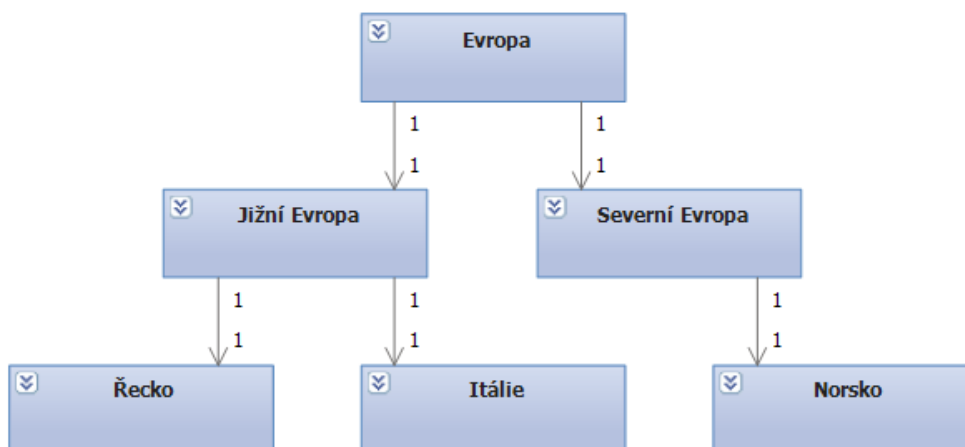
Datumových vlastností je v TARS více druhů.

- Přesné datum – podobnost je určena vzdáleností od zadaného data. Výpočet proto můžeme použít analogicky z kapitoly 5.4.3.
- Den v týdnu – Nabízí se výpočet pomocí modulo. Jde opět o podobnost založenou na vzdálenosti, jen je nutné myslet na to, že například neděle je stejně vzdálená od soboty jako od pondělí dalšího týdne.
- Měsíc v roce – Výpočet téměř nemá smysl. Pokud uživatel požaduje například dovolenou v lednu, ostatní měsíce jednoduše mají podobnost 0. Ke zvážení jsou snad jen nejbližší měsíce, kterým ovšem stačí nastavení pevné hodnoty podobnosti.

#### 5.4.5 Taxonomie

Taxonomii - stromové zatřídění hodnot využijeme například u destinací. Přináší to přirozené určení podobností. Při jejím získávání najdeme nejkratší cestu mezi pojmy a následně ji oceníme. To znamená, že sečteme všechny spojnice. Je možné nastavit cenu každé hrany a vypočítávat cenu trasy jemněji, my si prozatím vystačíme s hodnotou hrany 1.

Konkrétně na obrázku 20 je výsek taxonomie destinací s ohodnocením hran 1. Hodnota podobnosti Řecko – Itálie je 2, zatímco vzdálenost Řecko – Norsko je již 4.



Obrázek 20 - Posuzování podobnosti v taxonomickém stromu

### 5.4.6 Výčtové vlastnosti

Posledním typem vlastnosti, u kterého zjišťujeme podobnost, je výčtový typ (viz tabulka 4).

Modelovým typem atributu je stravování. Do jisté míry bychom vystačili i se vzdáleností položek, ale ty nemusí být zákonitě správně seřazeny, nebo se nemusí lišit stejně. Proto nastavujeme hodnoty v matici metodou každý s každým. Tak je totiž možné pokrýt také nesymetrickou verzi. Konkrétně, pokud uživatel požaduje (osa X) UAI (Ultra All Inclusive) pak předpokládáme, že se s AI (All Inclusive smíří) podobá na 70 %. Naopak bude-li požadovat AI, pak UAI je jistě příjemným bonusem a my získáváme podobnost 90 %.

Y ↓ X →	Vlastní	Snídaně	Večeře	Obědy	Polopenze	PP	AI	UAI
Vlastní	1,0	0,5	0,2	0,2	0,0	0,0	0,0	0,0
Snídaně	0,5	1,0	0,5	0,5	0,2	0,0	0,0	0,0
Večeře	0,2	0,5	1,0	0,8	0,6	0,2	0,0	0,0
Obědy	0,2	0,5	0,8	1,0	0,8	0,2	0,0	0,0
Polopenze	0,0	0,2	0,6	0,8	1,0	0,8	0,5	0,4
PP	0,0	0,0	0,2	0,2	0,8	1,0	0,8	0,6
AI	0,0	0,0	0,0	0,0	0,5	0,8	1,0	0,7
UAI	0,0	0,0	0,0	0,0	0,4	0,6	0,9	1,0

Tabulka 4 - Výčet podobností pro vlastnost typu enum – Stravování (PP – Plná penze, AI – All Inclusive, UAI – Ultra All Inclusive)

### 5.4.7 Normalizace

Některé podobnosti nemusí být po výpočtu v námi požadovaném tvaru. Například u číselných typů dostaneme jen vzdálenost dvou hodnot. Takové číslo musíme normalizovat, abychom jej převedli do intervalu 0-1. K tomu použijeme vzorec:

$$d(x) = (x - \min(x)) * \frac{1}{\max(x) - \min(x)}$$

x je hodnota kterou chceme normalizovat a min(x) a max(x) jsou hranice původního intervalu hodnot čísla x.

Nesmíme zapomenout na to, že výsledkem některých výpočtů není podobnost ale rozdílnost. Pak stačí použít následující vzorec:

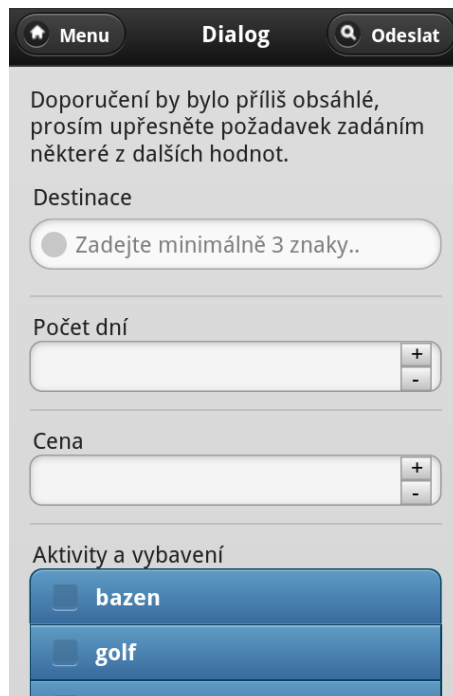
$$\text{sim}(x, y) = 1 - d(x, y)$$

## 5.5 GUI

GUI je z důvodů uvedených v kapitole 2 implementováno jako webové rozhraní přizpůsobené pro mobilní zařízení, konkrétně mobilní telefon. Vybraná platforma byla použita z důvodu rychlého pokrytí co největšího množství klientů, bez nutnosti instalace na straně uživatele. Z uživatelského rozhraní popíši jen části podstatné pro doporučování informací a dialog.

### 5.5.1 Zadání požadavku a dialog

je řešeno jako výpis seznamu parametrů (otázek v pořadí podle významnosti) podle vah jednotlivých vlastností. Grafické provedení je podobné dialogu (viz obrázek 21).



Obrázek 21 - GUI - Zadání požadavku/dialog

#### Dialog

Pokud systém vyhodnotí požadavek jako pod-specifikovaný, následuje dialog pro doplnění dalších parametrů k většímu omezení kandidátské sady nabídek. Grafické provedení viz obrázek 21. Zobrazeny jsou již jen ty atributy - otázky, na které uživatel ještě neodpovídal, a to v pořadí podle algoritmu uvedeného v kapitole 5.2.2.

Nabízí se varianta zobrazení jen jediné nejvhodnější otázky, to se ale neukázalo jako vhodné, protože uživatel není vždy schopen odpovědět na námi nabídnutou otázku. Proto je zachována nejširší možná nabídka otázek.

### 5.5.2 Prezentace výsledků doporučení

#### Vysvětlení doporučení

Při prezentaci nabídek je dobré myslet na vysvětlení doporučení. Zejména u systémů založených na metodě průběžných doporučení (viz 4.3.2) se bude uživatel mnohem lépe orientovat, pokud bude znát důvody, proč jsou mu doporučeny právě zobrazené nabídky. Dále se pak rozhoduje jak výsledky zpřesnit.

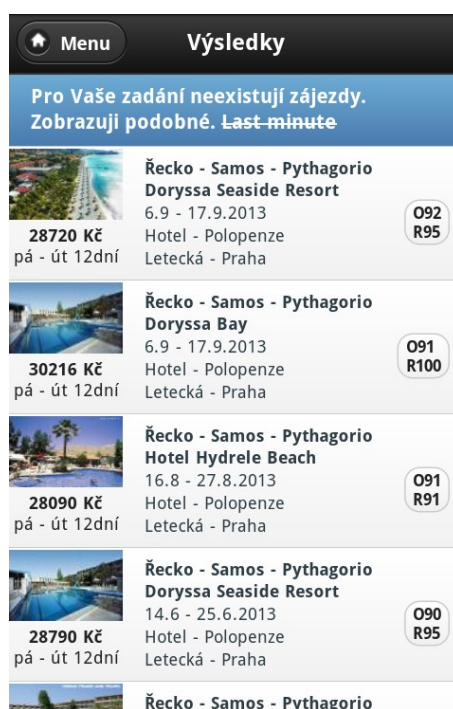
V prototypu systému byly jako orientační parametry zavedeny procentuální ohodnocení podobnosti (viz obrázek 22):



- podobnost nabídky vzhledem k požadavku – na obrázku v informační bublině označeno jako Rxx
- podobnost nabídky vzhledem k nabídce z referenční OC – na obrázku v informační bublině označeno jako Oxx

Podobnost nabídky vzhledem k požadavku je pouze orientační, protože neexistuje úplná shoda vlastností požadavku a nabídky. Informační bublina je zobrazena pouze volitelně.

Zde je prostor pro další vylepšení. Každé znázornění procentuální hodnoty by mohlo být doplněno vysvětlením, který z parametrů hrál hlavní roli v jeho výši. Například: Zájezd1 má vysoké hodnocení, protože má výhodnou cenu, naopak Zájezd4 má hodnocení nízké právě kvůli vysoké ceně. Otázkou zůstává, nakolik je taková informace uživatelsky zajímavá.



Obrázek 22 – GUI - Základní výpis seznamu doporučení

## Informace o uvolnění požadavků

Důležitým parametrem, který je nezbytný pro rozhodování uživatele, je informace o uvolnění vlastnosti požadavku. K takové situaci dochází při nezvládnutelném přespecifikovaném požadavku (viz kapitola 5.2.3). Na obrázku 22 je modelový případ, kdy uživatel odeslal dotaz a systém vyhodnotil, že na jeho základě neexistují nabídky k vrácení. Proto dojde k automatickému uvolnění vlastnosti požadavku, konkrétně Last minute. Uživatel je o této situaci informován textovým hlášením, doplněným přeškrtnutým seznamem parametrů, jež byly uvolněny.

## Různorodost odpovědí

V kapitole 4.3.2 (část Různorodost výsledků) jsem se již o různorodosti výsledků zmínil. Je dobré zajistit, aby výsledky doporučení nejen co nejvíce odpovídaly preferencím, ale také by měly

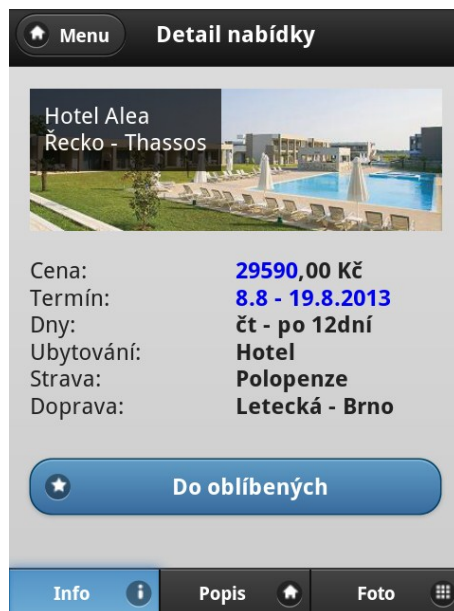
uživatelé dávat dostatečný manévrovací prostor pro rozhodnutí. V doméně zájezdů se velmi často stává, že je ve výsledcích hledání vypsané několik téměř stejných nabídek. Liší se jen velmi málo a pokud uživatel nemá zájem už o první z nich, je velmi pravděpodobné, že i ostatní mu vyhovovat nebudou.

Proto by další doporučené zájezdy měly mít vzájemnou odlišnost. Metody uvedené v kapitole 4.3.2 nejsou pro TARS vhodné. Bounded Greedy má pro svou výpočetní náročnost nedostatečný zisk a například hladinový výpis podle podobnosti nemá dostatečnou rozlišovací schopnost. Díky tomu, že jsou termíny často téměř stejné, zůstává v hladinách příliš mnoho nabídek. Proto jsem se rozhodl zajišťovat různorodost pomocí výpisu podle zájezdů nebo hotelů. Ty jsou pro termíny zastřešující vrstvou a ve většině případů nabízí dostatečně různé nabídky.

### 5.5.3 Prezentace nabídky – detail zájezdu

Po té, co uživatel vybere jednu z doporučených nabídek, dostane na obrazovku detailní informace o zájezdu – odpovídá detailu zájezdu (viz kapitola 3.4.4). Z důvodu velmi omezeného prostoru, bylo nutné mnohé informace skrýt a maximálně využít záložkového systému zobrazování. Z pohledu TARS je zde nejdůležitější položkou tlačítko nebo sada tlačítek zpětné vazby o užitečném doporučení. Jsou to funkce jako: odeslání nabídky na email, doporučení příteli, objednání zájezdu atp. Každá z těchto voleb znamená, že uživatel se maximálně ztotožňuje se zvolenou nabídkou a počítá s ní v dalším výběru, nebo je jeho finálním výběrem. To je pro systém důležitá informace, kterou si CBR uloží jako offercase.

Na obrázku 23 je detail nabídky, konkrétně zájezd do Řecka. V první záložce, která je zobrazena, jsou základní informace o zájezdu. Rozšířený popis a fotografie jsou dostupné na dalších záložkách. Nezbytným prvkem zpětné vazby je v prototypu pro testování tlačítko „Do oblíbených“. Tím mají uživatelé možnost uložit nabídku do seznamu oblíbených zájezdů a pro systém TARS je to informace o užitečném doporučení.



Obrázek 23 - Detail nabídky

### 5.5.4 Reprezentace případu CBR

V kapitole 4.2.3 jsem již nastínil standardizované možnosti popisu případu pomocí XML – CBML. Pro TARS jej nepovažuji za příliš vhodný, a to z důvodu vysoké režie XML, a to jak pro část parsování ze souborů tak pro udržení případů v paměti serveru. Nemluvě o poměrně nemalém nároku na diskový prostor po delší době užívání TARS. XML bohužel obsahuje mnoho redundantních dat. Tuto nevýhodu nevyváží ani otevřenost a znovu-použitelnost, protože není zamýšleno použít data z CBR mimo TARS bez jeho samotného zásahu. Pokud bude existovat zájem o využití dat v jiné části systému, TARS poskytne data již zpracovaná.

Výše uvedené problémy částečně eliminuji využitím objektů v paměti s úložištěm v databázi s pomocí repozitářů. Objekty uložené v databázi umožní také rychlejší vyhledávání v případě, že již nebudeme schopni udržet všechny případy v paměti RAM. Databáze urychlí vyhledání případů, které by bylo v podobě souborů XML nepřiměřeně zdlouhavé.

### 5.5.5 Reprezentace vlastností a míry podobnosti

Vzhledem k tomu, že jsou míry podobnosti velice citlivé na nastavení, pro jejich reprezentaci využiji XML formát. Ten umožní pozdější úpravy a doladování vah a samotného výpočtu podobnosti jednotlivých vlastností, bez nutnosti zasahovat do kódu aplikace. Otevřenost XML přináší možnost pro nastavení výše zmíněných parametrů později vytvořit GUI na jakékoli platformě, pro vysoký komfort práce doménového experta. Takový záměr bude vysoce vhodný ve spojení s množstvím vygenerovaných statistik atp.

Jak jsem již uvedl v kapitole 4.2.3, existuje standardizovaný postup popisu – CBML. Pro potřeby TARS jsem jej rozšířil a upravil. Ukázka konfiguračního souboru XML je uvedena na obrázku 24. Soubor již neobsahuje jen informace pro výpočet podobnosti nebo informace o vlastnosti, ale obojí bylo spojeno pro jednoduchost administrace.

```
<IntegerFeature Name="Počet dní" URL="dny" DSURL="Dod-Ddo" Weight="0.5">

  <Similarities>
    <Interval MinInclusive="0" MaxInclusive="1"
              MinSimilarity="1" MaxSimilarity="0.8" />
    <Interval MinInclusive="2" MaxInclusive="3"
              MinSimilarity="0.8" MaxSimilarity="0.3" />
    <Interval MinInclusive="4" MaxInclusive="128"
              MinSimilarity="0" MaxSimilarity="0" />
  </Similarities>
  <MinInclusive>0</MinInclusive>
  <MaxInclusive>128</MaxInclusive>
  <SimilarityType>symmetrical</SimilarityType>
</IntegerFeature>
```

Obrázek 24 - Konfigurační soubor pro nastavení vlastnosti

V ukázce je konkrétní nastavení vlastnosti „Počet dní“. Protože některé informace podléhají utajení, jsou zobrazeny jen ty nejdůležitější pro pochopení struktury a některé hodnoty byly pozměněny. Patrné jsou intervaly omezení hodnot vlastnosti, jejich reprezentace v URL a váha. Tag „Similarities“ nese nastavení nutné pro hladinový výpočet podobnosti.

## 6 Nastavení a výsledky

Systém TARS umožňuje velké množství nastavení převážně pro metriky podobnosti. Každý atribut má své váhy a stanovení míry podobnosti pro hladiny hodnot, ocenění cest taxonomie, odlišnosti hodnot apod., v závislosti na typu vlastnosti. Všechny tyto hodnoty byly nastaveny podle zkušeností doménového experta.

### 6.1 Nastavení

Váhy atributů jsou pak dále automaticky upravovány (viz kapitola 5.3.1). Mimo tato nastavení vlastností, TARS vyžaduje také určení míry inkrementace resp. dekrementace učení vah a prahových hodnot zvládnutelného setu.

#### 6.1.1 Nastavení prahových hodnot zvládnutelného setu

Horní mez zvládnutelného setu byla zavedena zejména kvůli omezení počtu zpracovávaných nabídek. Protože získání podobnosti je poměrně výpočetně náročné, závisí tato hodnota na výkonnosti serveru, který jej zpracovává. Pro zvolený testovací server se doba zpracování pohybovala kolem 1 sec na 12 000 porovnávaných nabídek. Doba pro zpracování byla omezena na 3 sec, a proto je doporučeno nastavení horní meze zvládnutelného setu pro aktuální server na 30 000 nabídek.

Ukázalo se, že dolní mez je vhodné nechat nastavenou na 0. Protože při dosti specifikovaném požadavku se stávalo, že v nabídce byl jen jeden možný zájezd, který byl také vhodný.

#### 6.1.2 Nastavení koeficientu inkrementace/dekrementace

Uživatelské váhy jsou upravovány podle chování uživatele. Využívány jsou převážně pro uvolňování atributů požadavku při jeho přespecifikování. Jde o dlouhodobou preferenci. Je ale třeba dbát na to, aby tato preference nebyla nepřekonatelná při dalším hledání v jiném kontextu. V tomto ohledu byl také stanoven interval pro růst váhy od 0 do 1. Při defaultním nastavení váhy 0.5 se osvědčil koeficient růstu 1,2. Ten zaručuje, že při dalším hledání v jiném kontextu stačí jen tři požadavky pro jeho korekci do původních výchozích hodnot podle vztahu:

$$0,864 * 1,2^3 = 0,5$$

Kde 0,864 je hodnota váhy po třech inkrementacích a také maximální dosažitelná hodnota v rámci stanoveného intervalu, 1,2 je hodnota koeficientu, 3 je počet kroků dekrementace a 0,5 je výchozí hodnota váhy.

### 6.2 Výsledky

Úspěšnost doporučování byla ověřena testováním na skupině uživatelů. Pro tento účel byly provedeny úpravy na API katalogu, systému TARS i nastavení. Abychom i s malým počtem hledání dosáhli měřitelných výsledků, bylo nutné omezit širší nabídky zájezdů. Konkrétně bylo v nabídce jen omezené množství destinací jižní Evropy.

V systému TARS byly objekty bází doplněny o statistické sloupce. Sledoval jsem počet kroků potřebných k dosažení úspěšného doporučení, datum a aktuální počty případů OfferCase i DialogCase.

Do testování byli zapojeni uživatelé znalí domény i uživatelé z řad široké veřejnosti. Osm zaměstnanců firmy převážně prodejců zájezdů reprezentovalo uživatele znalých domény. K nim bylo připojeno 6 klasických uživatelů.

Každý z uživatelů měl na svém nebo zapůjčeném zařízení provést 5 hledání v následujícím rozložení:

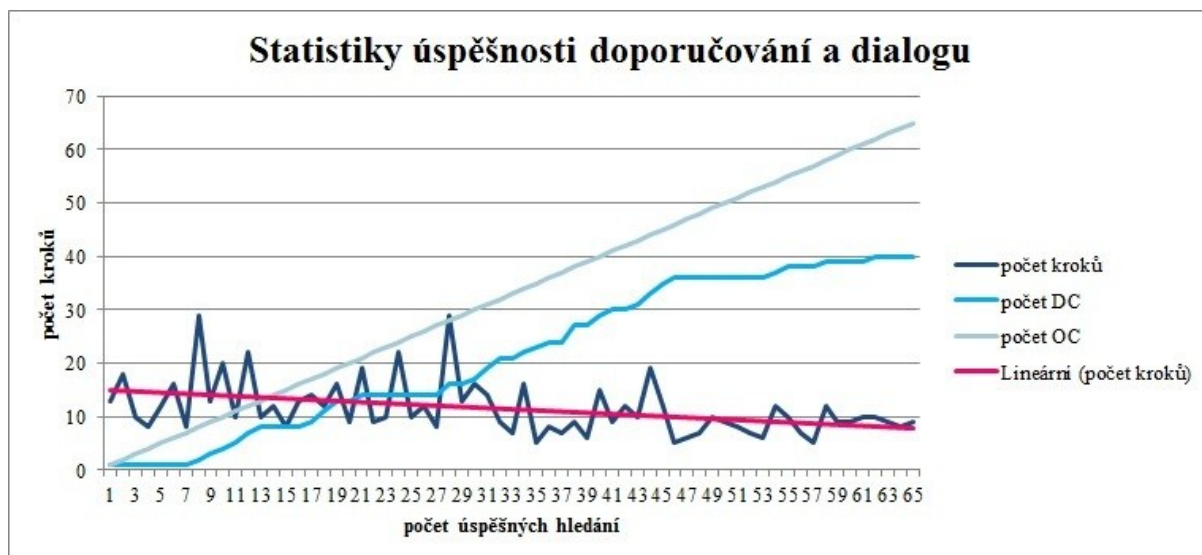
- 3 různá hledání zájezdů dle vlastního zájmu,
- jedno hledání záměrně pře-specifikované – s nutností vyplnit všechny atributy formuláře,
- jedno hledání záměrně pod-specifikované – s výběrem jen jednoho atributu s nízkou omezovací silou a následným dialogem.

Cílem testu bylo ověřit úspěšnost doporučování a dialogu v závislosti na čase a množství naučených případů. Jako ukazatel úspěšnosti byl vybrán počet kroků potřebných k úspěšnému doporučení. Předpokládáme, že všichni uživatelé testovacího vzorku zakončí hledání úspěchem. Rozhodující tedy je, kolik kroků k tomu potřebovali. Uživatel nespokojený s doporučením musí opakovat hledání a počet kroků roste. Stejně tak roste, i pokud uživatel není schopen rychle zadat svůj požadavek z důvodu špatného výběru otázek dialogu.

V rámci testu jsem se také zajímal o zkušenosti uživatelů s grafickým rozhraním, tedy přirozené orientaci a komfortu jeho užívání.

## 6.3 Statistiky

Před provedením testu byly vymazány všechny případy i statistické informace. Důvodem byla snaha zaznamenání výchozího stavu a problému studeného startu. Někteří uživatelé neprovedli všech 5 hledání a ze statistik byly také vyřazeny dva extrémy velkého množství kroků, způsobeného seznamováním se s uživatelským rozhraním. Úspěšných hledání zařazených do statistik bylo zaznamenáno 65. Průměrný počet kroků potřebných k dosažení úspěchu byl 11. Na grafu (viz obrázek 25) jsou zakresleny křivky počtu kroků, počet DialogCase a OfferCase. Graf byl doplněn o trend vývoje počtu kroků pro lepší názornost.



Obrázek 25 - Graf úspěšnosti doporučení a dialogu

Z grafu je znatelné, že počet kroků se mírně snižuje s přibývajícím počtem naučených případů. Při prvních hledáních nejsou viditelné žádné výrazné extrémy, problém studeného startu se tedy neprojevil. Počty DialogCase rostly ve většině případů u hledání s větším množstvím kroků, uživatelé tedy měli problém se zadáním a byl využíván dialog. Zajímavé je, že se neukázal rozdíl mezi obyčejnými uživateli a těmi, kteří ovládají doménu prodeje zájezdů.

Z uvedeného je patrné, že použité řešení CBR doporučení a sestavování dotazu s pomocí dialogu se stává jen mírně úspěšnějším s narůstajícím počtem naučených případů. Netrpí ale studeným startem, a proto by mělo být řešení vhodné u projektů se slabou zpětnou vazbou.

Uživatelé se v systému orientovali poměrně zdatně, jen měnící se seznam nabízených atributů (otázek) hodnotili jako rušivý. Problémem se ukázalo navržené řešení výběru destinace, které je ale spíše otázkou grafického zpracování.

## 7 Závěr

Cílem diplomové práce bylo vybrat metodu pro doporučování informací v mobilních zařízeních a experimentálně ji ověřit v doméně prodeje zájezdů. V průběhu práce byly popsány specifika mobilních zařízení, problémová doména, obecné způsoby doporučování informací a dialogu a prototypová aplikace TARS. Pro TARS (Travel Agency Recommender System) byla pro oblast dialogu i doporučení vybrána metoda CBR, zejména z důvodů orientace na kontext a eliminace problému studeného startu pro velké množství položek báze.

Úkolem systému TARS je doporučení zájezdu na základě potřeb a chování uživatele. Při explicitním sestavování požadavku pomáhá uživateli pomocí podpory dialogu a následně doporučí seznam nabídek. Výběr některé z nich je pro systém důležitým zdrojem zkušeností. Ke správné funkci systému je nutný sběr dlouhodobých i krátkodobých preferencí uživatele a zejména uchovávání úspěšných případů doporučení pro užití v budoucnu.

Testem bylo ověřeno, že pro problémovou doménu je vybraná metoda vhodná i bez dlouhodobého sběru informací uživatelského profilu a množstvím naučených případů CBR se mírně zlepšuje úspěšnost doporučování.

Byly navrženy možnosti vylepšení výpočtu metrik i sofistikovanějšího algoritmu uvolňování atributů požadavku. Také bude nutné upravit uživatelské rozhraní tak, aby změna seznamu otázek dialogu nepůsobila rušivě a celkově se tak zlepšila přirozená orientace v systému. Velký prostor je také v doladění nastavení metrik podobnosti, které jsou již plně závislé na zkušenostech doménového experta. Po doplnění objednávky je rozhodnuto o zkušebním nasazení systému TARS do produkčního prostředí.



# Bibliografie

1. McGill, Shaun. Definice smartphonu. *PdaSoft*. [Online] 2010. [Citace: 20. 7 2012.] <<http://www.pdasoft.cz/modules.php?name=News&file=print&sid=4848>>
2. SIVULKA, Ján. *Improvement of access to information on mobile devices*.
3. Doseděl, Tomáš. UMTS, HSPA+, LTE: vyznejte se v datových přenosech. [Online] 2012. [Citace: 29. 4 2013.] <<http://www.mobinfo.cz/umts-hspa-lte-vyznejte-se-v-datovych-prenosech/>>
4. Brownlow, Mark. Smartphone statistics and market share. *Email marketing reports*. [Online] 2012. <<http://www.email-marketing-reports.com/wireless-mobile/smartphone-statistics.htm>>
5. Tomasevic, Predrag. Windows Mobile, iPhone, Android - Marketplace Comparison. *Code Project*. [Online] 2010. [Citace: 20. 7 2012.] <<http://www.codeproject.com/Articles/51835/Windows-Mobile-iPhone-Android-Marketplace-Comparis>>
6. *Zákon o některých podmínkách podnikání v cestovním ruchu. 159/1999sb § 1 odstavec (1)*.
7. Slovníček pojmů. *Sopka*. [Online] [Citace: 25. 6 2012.] <[http://www.sopka.cz/slovnicek\\_index.php](http://www.sopka.cz/slovnicek_index.php)>
8. Vlček, Jan. Velké srovnání mobilních e-shopů. *iDnes*. [Online] 2012. [Citace: 20. 7 2012.] <[http://mobil.idnes.cz/srovnani-mobilnich-e-shopu-dd1-/aplikace.aspx?c=A120327\\_004545\\_aplikace\\_ham](http://mobil.idnes.cz/srovnani-mobilnich-e-shopu-dd1-/aplikace.aspx?c=A120327_004545_aplikace_ham)>
9. Burke, Robin. *Hybrid Recommender Systems: Survey and Experiments*. Fullerton : California State University.
10. Cvangroš, Petr. *Universal recommender system*. Charles University in Prague, 2011. str. 91.
11. Hiralall, Manisha. *Recommender systems for e-shops*. Amsterdam : Faculty of Sciences, 2011. str. 34.
12. Alexander Felfernig, Gerhard Friedrich, Dietmar Jannach. *Developing Constraint-based Recommenders*. In: *Recommender system handbook*. 2011. stránky 187-215.
13. Schank, Riesbeck and. *Inside Case-Based Reasoning*. Cambridge : Lawrence Erlbaum Associates, 1989.
14. Bergmann, Ralph. *Introduction to Case-Based Reasoning*. In: *Proceedings of the Third International Conference on Case-Based Reasoning*. 2000.
15. Aamodt a Plaza. *AI Communications*. IOS Press, 1994. stránky 35-59.

16. Lee, Danielle. *Case-based Recommendation*. 2011.
17. Smyth, Barry. *Case-Based Recommendation*. In: *The adaptive web*. Berlin : Springer-Verlag, 2007. stránky 342-376. ISBN:978-3-540-72078-2.
18. Ralph Bergmann, Janet Kolodner, Enric Plaza. *Representation in case-based reasoning*. In: *The Knowledge Engineering Review*. Cambridge : Cambridge University Press, 2005.
19. Lorcan Coyle, Conor Hayes, Pádraig Cunningham. *Representing Cases for CBR in XML*. Dublin : Trinity College. str. 10.
20. Ricci, Francesco. *Part 15: Knowledge-Based Recommender Systems*.
21. Skácel, Dalibor. Co to je entropie. *Tzb Info*. [Online] 2002. [Citace: 5. 3 2013.] <<http://www.tzb-info.cz/925-co-to-je-entropie>>
22. Doyle, Michelle a Cunningham, Pádraig. *A Dynamic Approach to Reducing Dialog in On-Line Decision Guides*. Dublin : Trinity College Dublin. str. 16.
23. Schmitt, Sasha. *simVar: A Similarity-Influenced Question Selection Criterion for e-Sales Dialogs*. In: *Artificial Intelligence Review*. Norwell : Kluwer Academic Publishers, 2002. stránky 195-221.
24. McCarthy, Kevin, Salem, Yasser a Smyth, Barry. *Experience-based critiquing : reusing critiquing experiences to improve conversation recommendation*. Dublin : University College, 2010. str. 15.
25. Barry Smyth, Paul McClave. *Similarity vs. Diversity*. In: *Case-Based Reasoning Research and Development*. 2001. stránky 347-361.
26. Mouzhi Ge, Fatih Gedikli, Dietmar Jannah. *Placing High-Diversity Items in Top-N Recommendation Lists*.
27. Doman, James. What is the definition of "personalization"? *Quora*. [Online] 19. 4 2012. [Citace: 1. 5 2013.] <<http://www.quora.com/What-is-the-definition-of-personalization>>
28. Dupont, Cedric. SearchWiki: make search your own. *Google Official Blog*. [Online] 20. 11 2008. [Citace: 28. 1 2013.] <<http://googleblog.blogspot.cz/2008/11/searchwiki-make-search-your-own.html>>
29. Stahl, Armin. *Combining Case-Based and Similarity-Based Product Recommendation*. In: *ECCBR'06 Proceedings of the 8th European conference on Advances in Case-Based Reasoning*. Berlin : Springer-Verlag, 2006. stránky 355-369. ISBN:3-540-36843-4 978-3-540-36843-4.
30. Lorcan Coyle, Padriag Cunningham. *Improving Recommendation Ranking by Learning Personal Feature Weights*.

31. Quinion, Michael. Gadget. [Online] 2007. [Citace: 25. 4 2013.]  
<<http://www.worldwidewords.org/qa/qa-gad1.htm>>

# Seznam obrázků

Obrázek 1 - Prodejnost podle operačního systému[comScore - únor 2012 [4]] .....	14
Obrázek 2 - Zjednodušený třídící diagram - Základní struktura DB v CK .....	17
Obrázek 3 - Formulář hledání zájezdů .....	20
Obrázek 4 - Položka výsledku hledání .....	21
Obrázek 5 - Náhled GUI aktuálního systému - Detail zájezdu .....	22
Obrázek 6 - Kolaborativní metody [Universal recommender system [10]] .....	24
Obrázek 7 – Metody založené na obsahu [Universal recommender system [10]] .....	25
Obrázek 8 – CBR [viz [14] podle [15]] .....	27
Obrázek 9 - Nesymetrická podobnost [Case based Recommendation - Barry Smith], .....	29
Obrázek 10 - Proces podobnosti případů [simVar] .....	32
Obrázek 11 - Obecný pohled na části systému TARS. ....	37
Obrázek 12 – Diagram aktivit – část dialogová .....	38
Obrázek 13 - Výběr seznamu otázek .....	40
Obrázek 14 – Uvolňování přespecifikovaného požadavku odstraněním atributů s nejmenší vahou ....	41
Obrázek 15 - Uvolňování požadavku úpravou hodnoty .....	41
Obrázek 16 – Zjednodušený třídící diagram - DialogCase .....	42
Obrázek 17 – Diagram aktivit – Průběh doporučování nabídek .....	44
Obrázek 18 - Class Diagram - OfferCase .....	46
Obrázek 19 - Graf lineární symetrické podobnosti .....	47
Obrázek 20 - Posuzování podobnosti v taxonomickém stromu .....	48
Obrázek 21 - GUI - Zadání požadavku/dialog .....	50
Obrázek 22 – GUI - Základní výpis seznamu doporučení .....	51
Obrázek 23 - Detail nabídky .....	52
Obrázek 24 - Konfigurační soubor pro nastavení vlastnosti .....	53
Obrázek 25 - Graf úspěšnosti doporučování a dialogu .....	56

# Seznam příloh

Příloha A: Náhledy GUI TARS

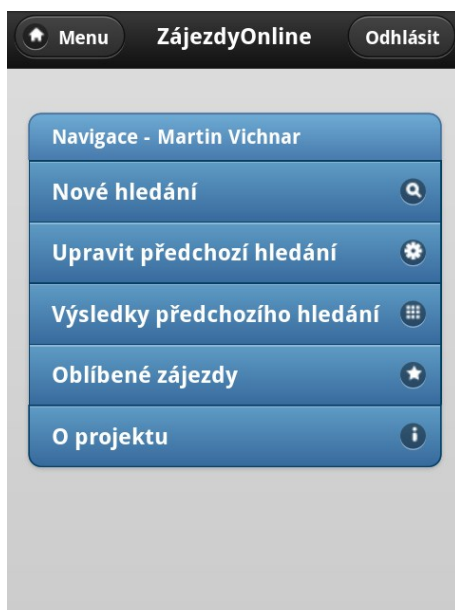
Součástí DP je CD.

## Obsah CD

Zdroje – obsahuje webové zdroje

DP.pdf – text diplomové práce včetně příloh

# Příloha A – Náhledy GUI TARS



*Obrázek 1 – Náhled systému TARS – Hlavní strana, rozcestník (přihlášený uživatel Martin Vichnar)*



*Obrázek 2 – Náhled systému TARS – Přihlašování pomocí rozhraní třetích stran*

Menu Požadavek Odeslat

Last minute

Last minute

Destinace

tha

Ithaka[Řecko]

Pythagorio[Řecko]

Thassos[Řecko]

Cena

Obrázek 3 – Náhled systému TARS - Požadavek (viz kapitola 5.5.2)

Menu Dialog Odeslat

Doporučení by bylo příliš obsáhlé,  
prosím upřesněte požadavek zadáním  
některé z dalších hodnot.

Destinace

Zadejte minimálně 3 znaky..

Počet dní

Cena

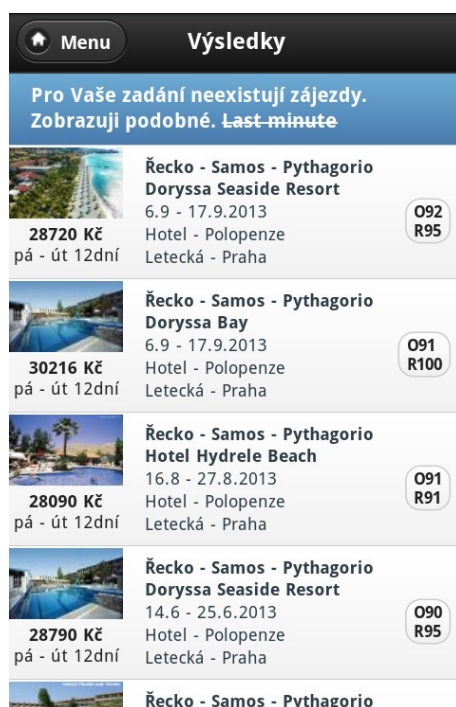
Aktivity a vybavení

bazen

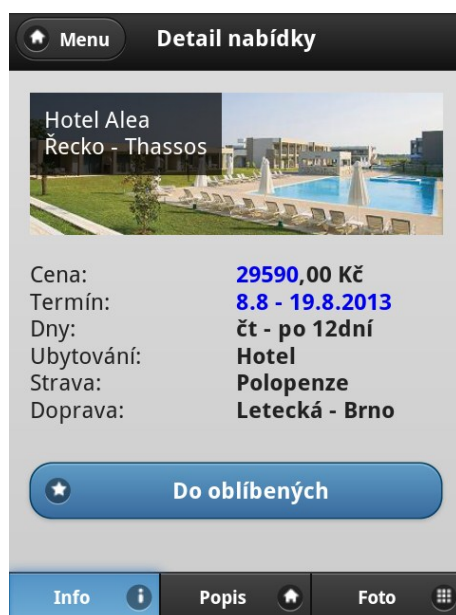
golf

Obrázek 4 - Náhled systému TARS - Dialog (viz kapitola 5.5.1)

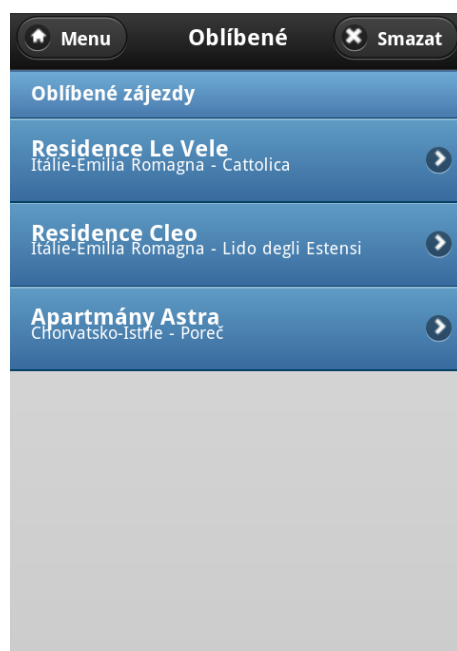




Obrázek 5 - Náhled systému TARS - Výsledky doporučení s uvolněným požadavkem Last minute (viz kapitola 5.5.2)



Obrázek 6 - Náhled systému TARS - Detail zájezdu (viz kapitola 5.5.3)



*Obrázek 7 - Náhled systému TARS - Seznam oblíbených zájezdů*